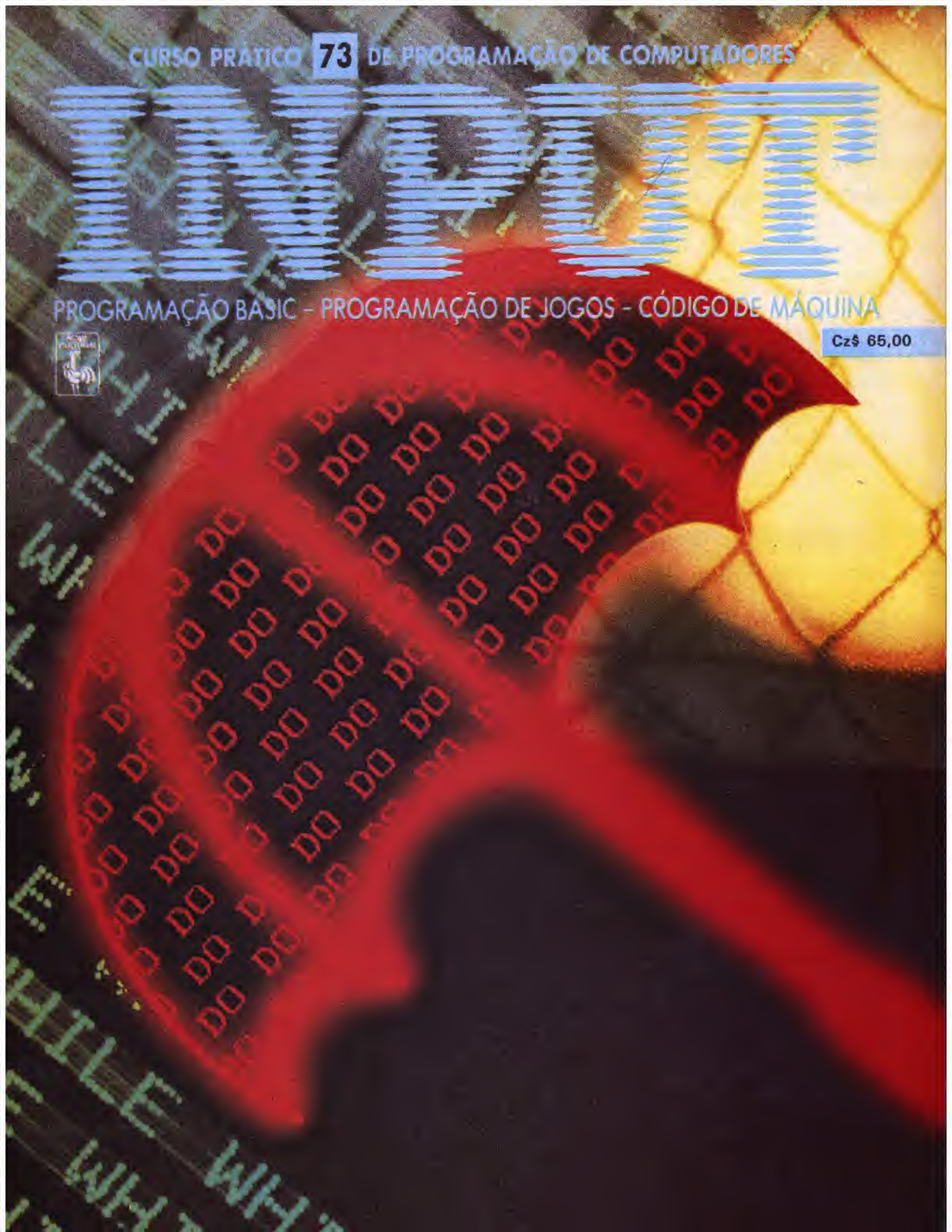
CURSO PRÁTICO 73 DE PROGRAMAÇÃO DE COMPUTADORES

BASIC II

PROGRAMAÇÃO BASIC - PROGRAMAÇÃO DE JOGOS - CÓDIGO DE MÁQUINA

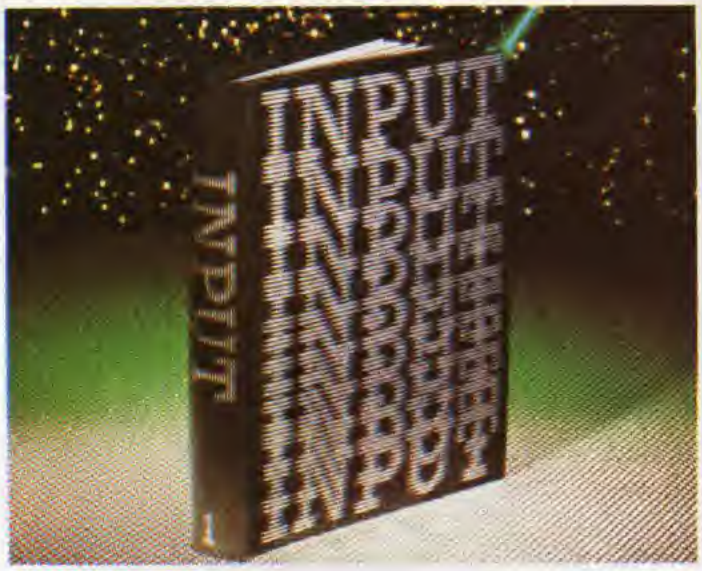
Cz$ 65,00



INPUT

Vol. 5 N.º 73

NESTE NÚMERO


INPUT
INPUT
INPUT

IMPRIMA SEUS DESENHOS

- TIPOS DE IMPRESSORA
- DESPEJO DE TELA
- PROGRAMAÇÃO GRÁFICA
- MONTAGEM DA TELA
- PROBLEMAS COM CORES

Ao desligar o micro, os maravilhosos desenhos que você criou na tela desaparecem sem deixar rastros. Com umas poucas linhas em BASIC, você poderá preservá-los em papel.

A capacidade gráfica dos micros permite a elaboração de vários tipos de gráfico ou desenho no vídeo. Quer você esteja interessado nas aplicações práticas desses gráficos, quer seja um artista preocupado em adotar um novo meio de expressão, as imagens produzidas com a ajuda do computador apresentam uma grande limitação: existem apenas enquanto a tela não for apagada. Se você quiser uma cópia permanente da imagem criada, precisará usar uma máquina fotográfica ou copiar a tela em uma impressora gráfica.

Para guardar uma listagem, uma lista de números ou um texto produzido por um programa, os usuários também têm que recorrer a uma impressora. Há, porém, uma grande diferença entre copiar uma tela de textos, escritos em ASCII padrão, e uma tela gráfica, de baixa ou alta resolução. Com exceção dos microcomputadores da linha Sinclair (ZX-81 e Spectrum), que possuem o comando **COPY**, os comandos de impressão do BASIC, como **LPRINT**, **PR#1**, **PRINT#-2** etc., funcionam apenas com caracteres ASCII. Alguns fabricantes oferecem impressoras que têm o conjunto ampliado de caracteres. Este inclui os caracteres semigráficos, típicos dos micros TRS-80, TRS-Color, TK-2000 e MSX, mas ainda assim não é suficiente para copiar uma tela com gráficos de média ou de alta resolução.

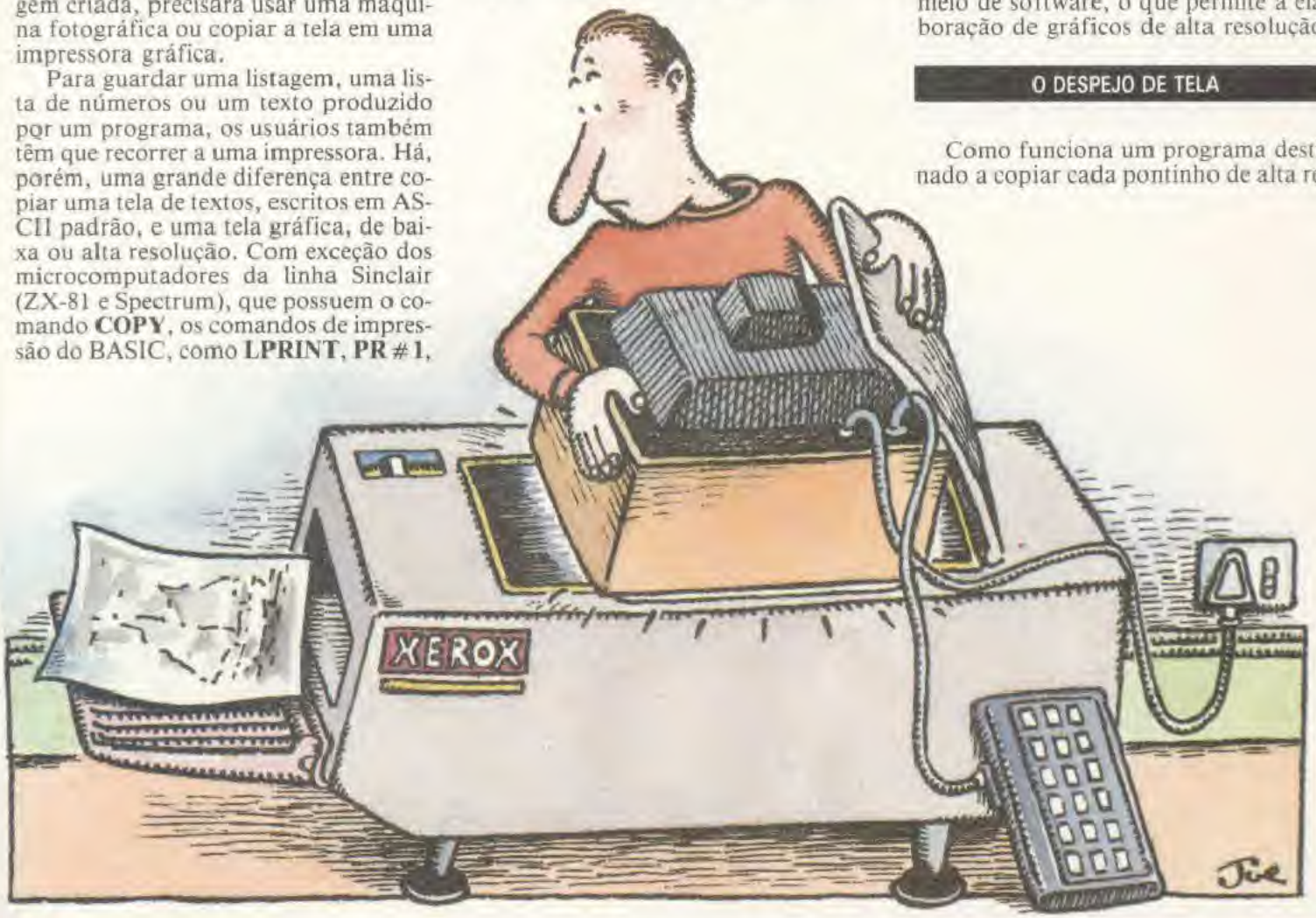
Como vimos em um artigo da seção *Periféricos* (página 648), existem dois modelos de impressora para micros: as impressoras de tipo formado (como as baseadas na "margarida"), e as impressoras matriciais. Embora algumas impressoras margarida possam fazer gráficos compostos de pontos, elas são muito lentas para ter alguma utilidade na cópia freqüente de telas gráficas.

As impressoras matriciais, por sua vez, imprimem os caracteres utilizando um conjunto de pontos, num processo idêntico ao da formação de caracteres na tela de vídeo. O sinal enviado pelo computador à impressora aciona uma série de pinos ou agulhas situados na cabeça de impressão, de modo a reproduzir um padrão de pontos que compõe um determinado caractere.

Nas impressoras matriciais não-gráficas (impressoras de texto), os padrões formados para cada caractere já estão pré-programados em uma memória ROM na impressora, e não podem ser manipulados ou alterados individualmente. Nas impressoras matriciais gráficas, porém, pode-se programar cada agulha da cabeça de impressão por meio de software, o que permite a elaboração de gráficos de alta resolução.

O DESPEJO DE TELA

Como funciona um programa destinado a copiar cada pontinho de alta re-



solução de uma tela gráfica? O comando **COPY** dos micros da linha Sinclair é um bom exemplo disso. Esse comando efetua uma operação que chamamos *despejo de tela* (*screen dump*, em inglês).

Com o micro acoplado a uma impressora adequada (própria para a linha Sinclair), o comando **COPY** "traduz" as configurações de pontinhos de cada setor da tela em padrões de atuação das agulhas da cabeça de impressão. Desse modo, o conteúdo da tela (texto e/ou gráfico) é fielmente reproduzido — menos a cor, evidentemente.

Os sistemas operacionais de alguns micros (como o TRS-80) podem ter funções de despejo de tela, que também só atuam se uma impressora compatível estiver acoplada à máquina. O CP-500 da Prológica, por exemplo, despeja a tela semigráfica na impressora P-500S, quando as teclas J, K e L são pressionadas simultaneamente no teclado.

A maioria dos micros, porém, não tem comandos ou funções especiais para produzir despejos de tela. E mesmo que seu computador disponha desses recursos, lembre-se de que eles nada produzirão sem a impressora adequada.

PROGRAMAÇÃO DA IMPRESSORA

Não é possível escrever na impressora usando o mesmo método com o qual escrevemos na tela. Se dermos, por exemplo, o comando:

```
PRINT CHR$(255)
```

no TRS-Color, um bloco colorido será exibido na tela (255 é o código do caractere gráfico correspondente). Entretanto, se você tentar fazer isso com a impressora, enviando o comando:

```
PRINT#-2,CHR$(255)
```

nada acontecerá, pois a impressora interpretará o código 255 de modo totalmente diferente. Dizemos, por essa razão, que os *códigos de controle* dos dois periféricos, vídeo e impressora, produzem efeitos diferentes.

Precisamos, assim, de um programa especial para efetuar o despejo de tela — um programa que prepare a impressora para receber gráficos. Algumas funções automáticas da impressora devem ser desligadas, e outras, ligadas. É necessário acionar a cabeça de impressão de forma que as agulhas possam ser controladas individualmente. Além disso, o avanço horizontal e vertical da cabeça de impressão precisa ser feito em passos diminutos, sem deixar espaços entre cada ponto impresso.

As impressoras gráficas podem ser programadas externamente para efetuar as funções mencionadas. Nesse *modo gráfico*, é possível atuar sobre grupos de agulhas usando códigos de oito bits. A informação captada na memória de vídeo é enviada para a impressora não sob a forma de códigos ASCII, mas como códigos binários correspondentes ao padrão de ativação das agulhas da cabeça. Assim, existem códigos binários para desligar o avanço automático de linha, para avançar a cabeça de uma linha para outra, para definir o espaço entre as linhas etc. O programa propriamente dito simplesmente "varre" a tela e produz as linhas de impressão correspondentes a cada grupo de bits.

Mais adiante, apresentaremos programas desse tipo para os micros Spectrum e TRS-Color. Antes disso, porém, precisamos examinar as combinações micro-impressora possíveis.

COMPATIBILIDADE

Os programas de despejo de tela dependem não só do tratamento dado à tela gráfica pelo computador, como também da marca e do modelo da impressora que será utilizada. Isso ocorre porque ainda não existe uma padronização dos caracteres de controle gráfico e da cabeça de impressão.

Nossos programas destinam-se às máquinas que adotam o padrão Epson. Este foi estabelecido para as impressoras do fabricante da marca Epson, e pode ser encontrado em modelos muito populares, como, por exemplo, o Epson MX-80, MX-120 e FX-80. No Brasil, várias empresas, entre elas a Elebra (impressoras Mônica e Alice), a Rima e a Grafix, seguem esse padrão.

As impressoras compatíveis com o padrão Epson usam uma matriz de oito pontos de altura. Esta é a configuração mais conveniente, pois permite ao micro enviar informações provenientes da tela, um byte de cada vez, pela interface paralela. O TRS-Color possui interface serial, mas o método é o mesmo. No Spectrum e no ZX-81, o usuário deve inserir uma interface paralela no conector de expansão. O Spectrum, além disso, precisa ser carregado com um programa em linguagem de máquina, para usar a impressora.

TEORIA DE PROGRAMAÇÃO GRÁFICA

A programação da seqüência de impressão das agulhas da cabeça para a obtenção de efeitos gráficos pode ser feita por meio de alguns comandos simples em BASIC. O processo é muito fácil. Para entendê-lo, devemos examinar como certos bytes enviados para a impressora são interpretados pelos circuitos de controle da mesma.

A impressora normalmente está capacitada para reconhecer os códigos ASCII e ASCII estendido, que estão pré-programados na memória ROM. Assim, se enviarmos uma seqüência de códigos entre hexadecimal 20 (correspondente a espaço em branco) e 7E (til), a impressora irá reconhecê-los como códigos de caracteres. Por exemplo, a palavra ABA é impressa quando enviamos:

```
LPRINT CHR$(65);CHR$(32);
CHR$(65)
```

Entre hexadecimal 00 e 1F situam-se caracteres de controle com diversas atribuições, que variam de impressora para impressora. No padrão Epson, **CHR$(7)** faz soar o alarme sonoro da impressora, **CHR$(13)** provoca um retorno de carro, enquanto **CHR$(12)** determina a mudança de página.

Outras funções de controle podem ser acionadas por meio da combinação de vários caracteres. Essas seqüências começam sempre pelo caractere **CHR$(27)**, que corresponde ao **ESCAPE** — por isso, são chamadas de *seqüências de escape*. A seqüência **ESC G**, por exemplo, coloca a impressora em modo de impressão em qualidade carta (cada caractere é impresso duas vezes):

```
LPRINT CHR$(27);"G"
```

Para mudar o espaçamento entre linhas de impressão, utiliza-se **ESC A n**, onde **n** refere-se ao número de passos por avanço de linha:

```
LPRINT CHR$(27);"A";CHR$(4)
```

Um passo é equivalente a 1/48 ou 1/72 de polegada, dependendo do modelo da impressora (com densidade simples ou densidade dupla). No exemplo anterior, o deslocamento entre cada linha de impressão equivale ao espaçamento entre quatro agulhas.

A seleção do modo gráfico, no padrão Epson, é feita pela seqüência **ESC K** ou **ESC L**. Quando você seleciona o modo gráfico, o código binário enviado à impressora aciona diretamente o padrão de agulhas da cabeça de impressão: quando o bit for 1, a agulha correspondente é acionada; quando o bit for 0, a agulha fica inativa. O bit menos significativo (D0) controla a agulha inferior, e o bit mais significativo (D7), a agulha superior. Assim, se quisermos imprimir uma coluna de pontos, a agulha de cima e as duas últimas agulhas de baixo não serão acionadas; o código a

enviar é 10000011, ou 128+0+0+0+0+0+2+1 = 131 em decimal.

A seqüência de escape precisa especificar, ainda, o número de bytes de código gráfico que serão enviados, e, em seguida, os bytes propriamente ditos: **ESC K n1 n2 dados**.

Os bytes **n1** e **n2** são, respectivamente, o mais significativo e o menos significativo de um número de dezesseis bits, e devem ser apresentados no conhecido formato **256 * n2+n1**.

Se vamos enviar 550 bytes gráficos, damos o comando:

```
LPRINT CHR$(27);CHR$(38);CHR$
(2);
```

pois 256*2 + 38 = 550. A expressão que se segue, em BASIC, serve para calcular **n1** e **n2** a partir de **n**, que representa o número total de bytes:

```
N1 = INT (N/256)
N2 = N-INT (N/256)
```

Este programa ilustra o padrão de acionamento das agulhas:

```
10 LPRINT CHR$(27);"A";CHR$(8)
20 LPRINT CHR$(27);"K";CHR$(16)
;CHR$(0);
30 FOR J=7 TO 0 STEP -1
40 LPRINT CHR$(2**J);
50 NEXT J
60 FOR J=0 TO 7
70 LPRINT CHR$(2**J);
80 NEXT J
90 LPRINT CHR$(10);
```

```
10 LPRINT CHR$(27);"A";CHR$(8)
20 FOR I=1 TO 5
25 LPRINT CHR$(27);"K";CHR$(16)
;CHR$(0);
30 FOR J=7 TO 0 STEP -1
40 LPRINT CHR$(2^J);
50 NEXT J
60 FOR J=7 TO 0 STEP -1
70 LPRINT CHR$(2^J);
80 NEXT J
90 LPRINT CHR$(10);
100 NEXT I
```

```
10 PRINT#-2,CHR$(27);"A";CHR$
(8)
20 PRINT#-2,CHR$(27);"K";CHR$
(16);CHR$(0);
30 FOR J=7 TO 0 STEP -1
40 PRINT#-2,CHR$(2^J);
50 NEXT J
60 FOR J=7 TO 0 STEP -1
70 PRINT#-2,CHR$(2^J);
80 NEXT J
90 PRINT#-2,CHR$(10);
```

```
5 PR#1
10 PRINT CHR$(27);"A";CHR$(8)
20 PRINT CHR$(27);"K";CHR$(16);
CHR$(0);
30 FOR J=7 TO 0 STEP -1
40 PRINT CHR$(2^J);
50 NEXT J
60 FOR J=7 TO 0 STEP -1
70 PRINT CHR$(2^J);
80 NEXT J
90 PRINT CHR$(10);
100 PR#0
```

A linha 10 estabelece um espaçamento de oito passos entre linhas. A 20 programa a impressora para o modo gráfico, avisando que a seguir serão enviados dezesseis bytes gráficos. Quando os dados não estão incluídos imediatamente na seqüência de escape, o ponto e vírgula no final dessa linha é obrigatório. Os laços das linhas 30 a 50 e 60 a 80 imprimem, respectivamente, um padrão descendente e outro ascendente de pontos (determinados pelos códigos que correspondem a potências de 2: 128, 64, 32 etc.). Finalmente, a linha 90 alimenta uma linha.

MONTAGEM DA TELA

Para copiar um desenho na impressora, é necessário executar um programa que o trace na tela. Existem duas alternativas para isso. Uma delas consiste em carregar o programa de desenho no computador e, em seguida, combiná-lo com o programa de despejo (a numeração das linhas de ambos tem que ser diferente). O primeiro programa deve ser alterado na tela depois que o desenho estiver pronto, de modo que o programa de despejo seja chamado automaticamente, ou quando se pressionar determinada tecla. Se achar conveniente, poderá incluir permanentemente o programa de despejo no programa de desenho, na forma de uma sub-rotina.

A segunda alternativa é a de mais fácil execução, mas depende da capacidade do micro de armazenar o conteúdo de telas gráficas em disco ou fita (comandos **BSAVE**, **SAVEM** etc.). Se seu computador tem essa capacidade, carregue e execute o programa de desenho e armazene a tela resultante em fita ou disco. Depois, carregue o programa de despejo usando um comando na linha 10 (essa linha não foi colocada nos programas que se seguem justamente para criar um espaço para o seu comando de carregamento). Com esse procedimento, o programa irá ler, em primeiro lugar, a tela armazenada.

Para os usuários do TRS-Color, o sistema é um pouco diferente: algumas linhas do programa de despejo têm que ser executadas antes que o desenho seja tracado na tela.

IMAGEM DE LADO

A imagem despejada por este programa é imprimida de lado — ou seja, no sentido longitudinal e não transversal, relativamente à imagem que aparece na tela. Com isso, obtemos uma imagem maior e mais margem no papel.

```
15 LPRINT CHR$5;
20 LPRINT CHR$27; "A"; CHR$8;
30 FOR x=0 TO 255 STEP 4
40 LPRINT CHR$13; CHR$27; "K";
CHR$0; CHR$96; CHR$1;
50 FOR y=0 TO 175 STEP 8
60 FOR d=0 TO -7 STEP -1
70 LET bt=(POINT(x,y-d)*128)+(P
OINT(x,y-d)*64)+(POINT(x+1,y-d)
*32)+(POINT(x+1,y-d)*16)+(POINT
(x+2,y-d)*8)+(POINT(x+2,y-d)*4)
+(POINT(x+3,y-d)*2)+(POINT(x+3,
y-d))
72 LPRINT CHR$bt;
74 LPRINT CHR$bt;
80 NEXT d
90 NEXT y
100 NEXT x
110 LPRINT CHR$4; STOP
```

A linha 15 desliga o conjunto de caracteres ASCII na impressora. O comando **LPRINT** da linha 20 informa à impressora que não deve haver espaçamento entre as linhas impressas.

O laço que vai da linha 30 à 100 varre uma linha da tela, em blocos de quatro pixels de cada vez. A linha 40 passa para a impressora um código de retorno de carro (código 13) e, em seguida, o código *0, que se encarrega de preparar a máquina para receber o número de bytes que a linha corrente de tela irá enviar. Esse byte é composto dos códigos 96 e 1 (também na linha 40), que são interpretados como 1*256+96 — ou seja, 352 bytes.

O laço da linha 50 à linha 90 percorre a tela de cima para baixo; o laço da linha 60 à 80 toma um bloco de oito pixels de cada vez. Antes de incrementar os laços, a linha 70 usa o comando **POINT** para ler os pixels na tela. Estes são condensados pela expressão numérica no byte **bt**, que é enviado duas vezes à impressora, pelas linhas 72 e 74. A segunda impressão fará a imagem aparecer duas vezes maior que na tela, na direção horizontal.

Finalmente, a linha 110 retorna a impressora ao conjunto ASCII e encerra

a execução. Se você quiser transformar esse programa em sub-rotina, não se esqueça de substituir o **STOP** dessa linha por um **RETURN**.

T

```
2 DIM A(8,1)
4 FOR K=0 TO 8: READ A(K,0),A(K
,1):NEXT
6 DATA 3,3,2,1,0,0,3,1,3,3,0,0,
1,2,3,1,3,3
20 PRINT #-2, CHR$(27);"A";CHR$
(8)
30 FOR L=0 TO 255 STEP 4
40 PRINT #-2,CHR$(13);CHR$(27);
"K";CHR$(0);CHR$(128);CHR$(1);
50 FOR K=191 TO 0 STEP -1
60 T=0:S=0:FOR M=0 TO 3:P=PPOIN
T(L+M,K):T=T*4+A(P,0):S=S*4+A(P
,1):NEXT
70 PRINT #-2,CHR$(T);CHR$(S);
80 NEXT K,L
90 PRINT #-2,CHR$(27);"@"
```

É importante que as três primeiras linhas do programa (2, 4 e 6) precedam a rotina de geração da tela gráfica que será impressa. Na linha 10, reservada para o seu programa, você deve estabelecer o tipo de **SCREEN** e de **PMODE**. Em virtude da forma de armazenagem de telas gráficas no TRS-Color, o programa de despejo precisa levar em conta as cores empregadas.

A linha 2 dimensiona um conjunto **A**, para armazenar o padrão de pontos usado em cada cor. Existem nove cores, mas nem todas podem aparecer na tela ao mesmo tempo. Os padrões estão definidos em **DATA**, na linha 6, e são carregados no conjunto **A** pela linha 4. Dois itens em **DATA** determinam cada cor; o segundo é impresso sobre o primeiro. Por exemplo, o primeiro par de números (3,3) indica preto, formando um padrão binário de 3 (11) sobre 3 (11). O segundo par (2,1) especifica verde, formando um padrão de 1 (01) sobre 2 (10). Note que há repetição de pares na seqüência, mas isso não importa: a segunda ocorrência determina uma cor que não pode aparecer na tela junto com a primeira especificação.

A linha 20 prepara a impressora para receber gráficos. A seqüência de escape que contém instrui a impressora a não dar espaçamento entre linhas.

O laço que vai da linha 30 à 80 percorre uma linha da tela, em blocos de quatro pixels. A linha 40 envia à impressora um código de retorno de carro (13) e, em seguida, o código *, que coloca a impressora em modo gráfico. Ainda nessa linha, **CHR$(0)** define a densidade de impressão e **CHR$(128);CHR$(1)** diz à impressora para esperar 128 + 1*256 bytes gráficos, ou seja, um total de 384 bytes.

O laço da linha 50 à 80 percorre a tela de cima para baixo (192 pixels de altura); o da linha 60 toma quatro pixels de cada vez, examina-os com **PPOINT**, calcula as cores e coloca o resultado nos bytes **S** e **P**. A linha 70 imprime esses bytes. Finalmente, a linha 990 reinicializa a impressora.

PROBLEMAS COM CORES

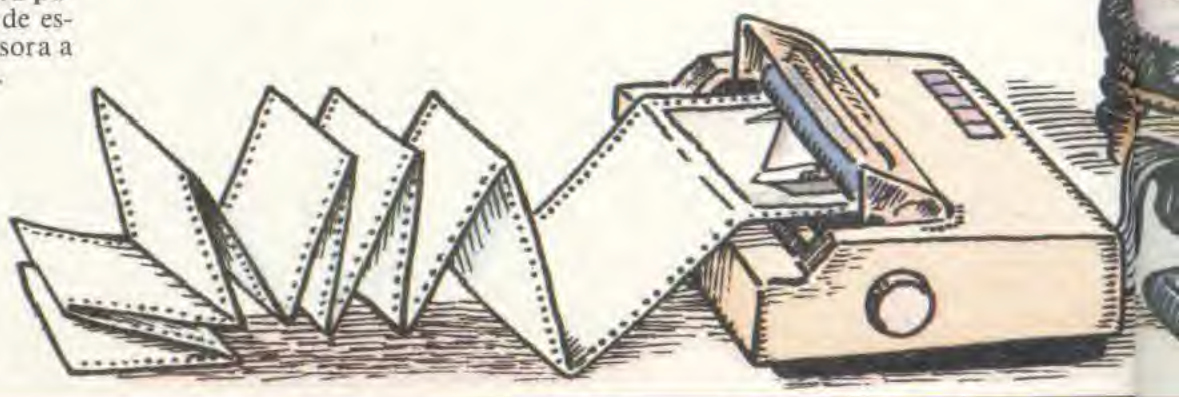
O programa de despejo de tela, em BASIC, imprime apenas uma representação binária, em preto e branco, da tela de alta resolução, pois reproduz o padrão de pixels sem diferenciar as cores presentes na imagem original. Em conseqüência, a imagem obtida apresenta densidade igual para todas as cores, não representando com fidelidade o desenho copiado.

É possível, porém, escrever um programa de despejo de tela em que as cores sejam representadas por diferentes tonalidades de cinza e preto. Esse efeito é conseguido por impressões sobrepostas de partes da imagem, de modo a obter tons mais escuros ou claros de cinza, conforme a cor presente. Em uma tela com quatro cores, por exemplo, podem-se definir quatro diferentes padrões de pontos, para representar o vermelho, com um cinza bem claro; o verde, com um cinza mais escuro etc. Isso é feito inibindo-se o avanço da cabeça de impressão.

Evidentemente, o tempo total de impressão é muito longo, nesse caso. Um despejo de tela simples demora quase meia hora em alguns micros; se levarmos em consideração a cor, esse tempo será aumentado para várias horas! Um programa em código de máquina — como o que apresentamos a seguir para duas linhas de computadores — pode resolver esse problema.

S

```
10 CLEAR 59999
20 LET L=100: RESTORE L: FOR N=
60000 TO 60247 STEP 8
30 LET T=0:FOR M=0 TO 7
40 READ A:LET T=T+A:POKE N+M,A:
NEXT M
50 READ A:IF A<>T THEN PRINT "E
RRO NOS DADOS DA LINHA ";L: STO
P
60 LET L=L+10: NEXT N : STOP
100 DATA 243,62,3,205,1,22,33,7
0,639
110 DATA 235,6,4,205,9,235,62,0
,756
120 DATA 50,83,235,62,0,50,84,2
35,799
130 DATA 6,175,221,33,85,235,19
7,62,1014
140 DATA4,237,75,83,235,245,205
,15,1099
150 DATA 235,245,205,30,235,241
,126,32,1349
160 DATA 6,203,63,203,63,203,63
,230,1034
170 DATA 7,214,7,237,68,221,119
,0,873
180 DATA 221,35,241,12,61,32,22
2,58,882
190 DATA 84,235,660,50,84,235,1
93,16,957
200 DATA 205,6,7,197,33,74,235,
6,763
210 DATA 9,205,9,235,221,33,85,
235,1032
220 DATA 6,175,197,6,4,30,0,197
,615
230 DATA203,35,203,35,221,126,0
,254,1077
240 DATA 0,40,7,221,53,0,62,3,3
86
250 DATA 24,2,62,0,131,95,221,3
5,570
260 DATA 193,166,228,123,245,21
5,241,215,1476
```



```
270 DATA 193,16,215,193,16,197,
33,65,928
280 DATA 235,6,5,205,9,235,62,1
0,767
290 DATA 215,58,83,235,198,4,50
,83,926
300 DATA 235,210,115,234,62,4,2
15,251,1326
310 DATA 201,126,35,215,16,251,
201,197,1242
320 DATA 205,170,34,71,4,126,20
3,7,820
330 DATA 16,252,230,1,193,201,1
97,62,1152
340 DATA 175,144,230,248,71,88,
22,0,978
350 DATA 203,35,203,18,203,35,2
03,18,918
360 DATA 121,203,63,203,63,203,
63,111,1030
370 DATA 38,0,25,17,0,88,25,193
,386
380 DATA 201,27,64,27,65,8,5,27
,424
390 DATA 65,8,27,65,0,13,27,42,
247
400 DATA 0,96,1,0,0,48,48,193,3
86
```

T

```
10 CLEAR 200,29992
20 CLS:FOR K=0 TO 12:T=0:FOR L=
0 TO 23:READ A
30 POKE 29993+24*K+L,A:T=T+A
40 NEXT:READ A:IF A<>T THEN PRI
NT "ERRO NOS DADOS DA LINHA";10
00+10*K:END
50 NEXT
1000 DATA 0,0,0,3,27,51,24,134,
254,151,111,111,140,242,111,140
,240,150,182,133,1,39,3,108,235
5
1010 DATA 140,231,77,39,4,129,2
,38,3,108,140,220,150,193,68,16
7,140,216,48,140,214,23,0,142,2
632
1020 DATA 95,52,4,51,141,1,1,19
8,191,166,228,52,6,134,4,52,2,2
3,0,134,48,140,107,166,1996
1030 DATA 134,167,192,108,97,10
6,228,38,240,53,2,53,6,166,228,
90,193,255,38,223,198,3,52,6,28
76
1040 DATA 51,141,0,212,198,192,
231,228,48,140,65,141,81,79,198
,4,72,72,106,192,43,2,138,3,263
7
1050 DATA 90,38,245,173,159,160
,2,173,159,160,2,106,228,38,230
,134,13,173,159,160,2,106,97,38
,2845
1060 DATA 207,53,6,134,10,173,1
59,160,2,53,4,173,159,160,0,129
,3,39,4,203,4,38,138,48,2059
1070 DATA 140,17,32,18,5,27,42,
4,128,1,3,1,0,2,3,0,1,2,3,2,27,
64,230,128,880
1080 DATA 166,128,173,159,160,2
,90,38,247,57,134,32,109,141,25
5,48,39,1,68,52,2,166,101,214,2
582
1090 DATA 182,193,1,34,1,68,230
,224,61,211,186,31,1,230,99,84,
84,84,109,141,255,18,39,1,2567
1100 DATA 84,58,166,99,109,141,
255,8,39,8,132,15,64,139,15,68,
32,5,132,7,64,139,7,198,1984
1110 DATA 1,74,43,3,88,32,250,1
09,141,254,238,39,14,52,4,197,8
5,39,5,88,235,224,32,3,2250
1120 DATA 84,235,224,52,4,166,1
32,164,224,84,37,3,68,32,250,17
1,141,254,206,171,141,254,203,5
7,3357
```

Embora o código de máquina esteja contido em uma lista de números em **DATA**, cada linha inclui somas de verificação, cuja finalidade é evitar que se cometam erros de cópia.

Não se esqueça de observar as precauções usuais ao rodar um programa em linguagem de máquina: para não correr riscos, armazene o programa, assim como suas versões corrigidas, em fita ou disco, antes de executá-lo.

Depois de testar o programa completo, rode-o com **RUN** e, em seguida, apague-o (o programa em código de máquina ficará armazenado em uma porção protegida da memória RAM). Digite ou carregue seu programa de desenho e execute-o. Para imprimir a imagem na tela, chame este comando:

S

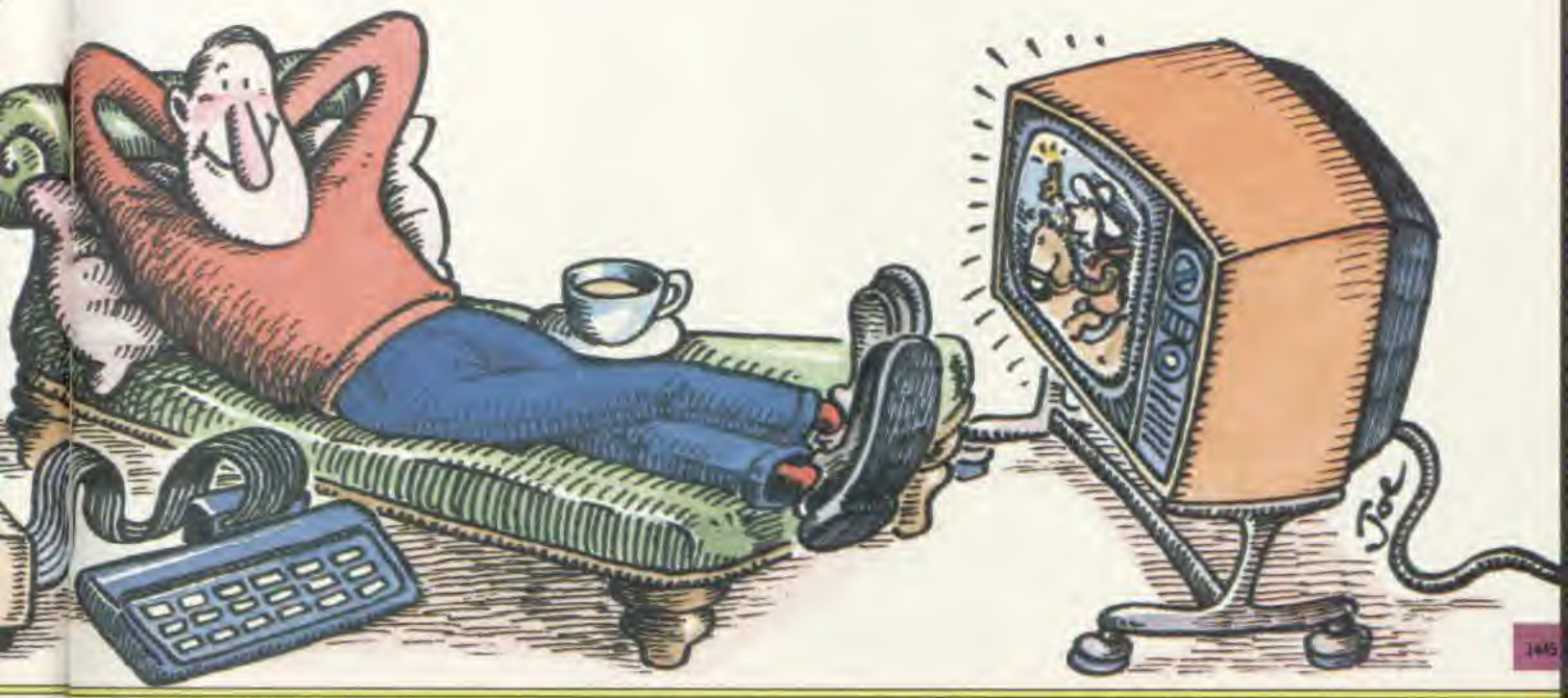
```
RANDOMIZE USR 60000
```

T

```
EXEC 30000
```

É importante desligar a alimentação automática de linha em sua impressora, antes de usá-la para despejos de tela. Para isso, siga as instruções do manual de operação.

Imprima a tela em uma fita de impressão já usada, pois isso proporcionará um contraste melhor entre as diferentes tonalidades.



ESTRUTURAS DO PASCAL

Já examinamos a filosofia e os princípios básicos do Pascal. Mostraremos agora como as estruturas dessa linguagem são organizadas na construção de um programa.

Como vimos no artigo da página 1436, é fundamental estudar a maneira de solucionar determinado problema antes de iniciar a elaboração de um programa em Pascal. O algoritmo para a resolução do problema deve ser aperfeiçoado e refinado até se tornar semelhante a um programa que possa ser compreendido pelo microcomputador.

Para realizar essa tarefa, você precisará conhecer as ferramentas disponíveis no seu micro, ou seja, as estruturas e os comandos aceitos pelo Pascal. Esse procedimento não é muito diferente daquele que adotamos ao elaborar um programa em BASIC. A partir de nossa experiência com a máquina, sabemos quais comandos são válidos, o que nos permite esboçar uma solução adaptável ao microcomputador.

No BASIC, diversos recursos podem ser utilizados em cada etapa da resolução de um problema. Esses recursos não se resumem a simples comandos, mas a combinações de vários deles. Por exemplo, quando precisamos repetir uma tarefa várias vezes, um laço **FOR...NEXT** costuma ser uma boa saída; para uma tomada de decisão, basta um **IF...THEN**. Existem estruturas semelhantes em Pascal — mas elas exigem um grau de refinamento maior que no BASIC. Neste artigo, examinaremos as mais simples e úteis dessas estruturas.

COMPATIBILIDADE

Ao contrário do BASIC, o Pascal é definido muito precisamente, não tendo variações entre um sistema e outro (como as existentes entre o BASIC do MSX e o do TK-2000, por exemplo). Assim, os programas que se seguem funcionarão em qualquer máquina cujo compilador aceite letras minúsculas.

Pela mesma razão, ao escrever seus próprios programas, você deverá usar uma determinada forma para cada operação. Para apresentar o padrão utilizado na definição da forma e da sintaxe do programa, pode-se recorrer a uma notação desenvolvida durante os anos 60, conhecida como Forma de Backus-Naur (BNF), ou ao diagrama de sintaxe.

Ambas as notações simplesmente mostram a estrutura geral de uma declaração. Suponhamos que se queira definir um identificador: *um identificador é definido como uma letra ou um dígito.* Em um diagrama de sintaxe, teríamos:

- COMPATIBILIDADE
- TÉCNICAS DE REPETIÇÃO
- WHILE...DO
- REPEAT...UNTIL
- FOR
- TOMADA DE DECISÕES
- IF...THEN...ELSE
- CASE
- DESENVOLVIMENTO DE UM PROGRAMA

Na notação de Backus-Naur, a mesma definição teria a seguinte forma:

<identific>::=<letra> | <letra ou digito>

Cada uma dessas representações mostra o significado exato da definição dada anteriormente. Para definir *letra*, você poderia escrever em BNF:

<letra>::=A | B | C | D | E | F | G etc.

Ou seja: uma letra é definida como A ou B ou C ou D ou E ou F etc.

As próximas estruturas-padrão serão dadas em BNF, para mostrar sua forma original, mas você reconhecerá facilmente as equivalentes em BASIC.

TÉCNICAS DE REPETIÇÃO

Existem três técnicas de repetição no Pascal. A escolha de uma delas depende do parâmetro a ser empregado no controle do laço. A primeira é familiar aos usuários do BASIC: **FOR...DO**. As outras duas, que também podem ser simuladas naquela linguagem, são: **WHILE... DO** e **REPEAT...UNTIL**.

A principal diferença entre essas técnicas está na condição de controle responsável pelo número de repetições. Na forma **WHILE...DO** (ENQUANTO...FAÇA), as instruções seguintes serão executadas enquanto determinada expressão continuar sendo verdadeira. A forma **REPEAT...UNTIL** (REPITA...ATÉ QUE) irá executar uma série de instruções até que uma dada condição seja alcançada. Ambas as técnicas são usadas quando não se sabe de antemão o número de repetições necessárias. Caso esse número possa ser previamente definido, utiliza-se a forma **FOR...DO.**

WHILE...DO

Em BNF, escreve-se:

while <expressão lógica> **do** <instrução>

Esse laço é empregado quando o número de repetições depende de uma condição. Assim, será executado enquanto a condição especificada no programa for verdadeira. Para repetir uma série de instruções dentro desse laço, adicione, após o **do**, um **begin** — indicando o início das instruções — e um **end** — indicando o término do laço.

O exemplo que se segue mostra o uso do **WHILE...DO**. Tente escrever o algoritmo em que o programa se baseia.

```
program exemplo2;
var no,sum:integer;
begin
   sum:=0;
   read(no);
   while no<>0 do
   begin
      sum:=sum+no;
      read(no)
   end;
   writeln(sum)
end.
```

REPEAT...UNTIL

repeat <instrução> **until** <expressão lógica>

As instruções entre **repeat** e **until** são repetidas até que a condição final se torne verdadeira. Todas elas são executadas pelo menos uma vez, pois o teste só é feito ao final. Ao contrário da forma **while...do**, não é necessário acrescentar **begin** e **end** a um conjunto de instruções. Usando-se essa técnica, o programa anterior pode ser escrito da seguinte maneira:

```
program exemplo3;
var no,sum:integer;
begin
   sum:=0;
   repeat
   read(no);
   sum:=sum+no;
until no=0;
write(sum)
end
```

Esse programa é mais eficiente que o anterior, pois, com o uso de **repeat...until**, economiza-se uma instrução **read(no)**. Para efetuar uma soma corrente, é melhor que o programa entre diretamente no laço inicializado por **repeat**. No programa **exemplo2**, o programa chega à solução por intermédio de uma estrutura menos adequada que a utilizada no **exemplo3**.

FOR

No Pascal, ao contrário do que acontece no BASIC, o laço inicializado por **FOR** só pode caminhar em passos (**STEP**) de +1 ou −1. Adaptando o programa anterior ao uso dessa técnica, teremos que introduzir inicialmente a quantidade de números a serem somados.

```
program exemplo4;
var no,sum,quant,i:integer;
begin
   read(quant);
   sum:=0;
   for i:=1 to quant do
   begin
      read(no);
      sum:=sum+no
   end;
   writeln(sum)
end.
```

TOMANDO DECISÕES

Como o BASIC, o Pascal dispõe de vários recursos para verificar qual decisão deve ser tomada diante de determinado resultado. O primeiro deles, comum ao BASIC, é o **if...then...else**, geralmente usado quando a seleção depende de um ou outro de dois eventos. Porém, se a decisão depende de um número maior de eventos, costuma-se utilizar **case...of**. Não há nenhuma estrutura similar a **case** no BASIC.

IF...THEN...ELSE

if <expressão lógica> **then** <instrução 1> **else** <instrução 2>

Essa estrutura indica que, se a expressão lógica for verdadeira, o programa executará a instrução 1; se for falsa, a instrução 2. Como mostra o diagrama FBN, o **else** é opcional; uma construção alternativa seria **if...then.**

CASE

A declaração **case** permite ao programa selecionar uma instrução de uma lista de várias possibilidades.

A forma geral de **case** é:

case expressão **of**

c1 : instrução;
c2 : instrução;
''
''
''
cn : instrução
end;

O valor da expressão deve corresponder a um dos **c** e ser do mesmo tipo. No Pascal padrão, se o valor da expressão não for encontrado na lista, uma mensagem de erro será exibida. Em algumas versões, porém, o programa simplesmente ignorará o **case**.

Note que há um **end** associado ao **case** que não corresponde a nenhum **begin**. Esse tipo de estrutura não é freqüente, e pode trazer inconvenientes durante a busca de um erro dentro do programa. Geralmente, a primeira coisa que se faz é verificar se há correspondência entre o número de **begin** e **end**. A associação do **end** ao **case** provocaria uma diferença. Para contornar o problema, coloca-se um rótulo em cada **end** — por exemplo, **end;** {**case**}.

As instruções equivalentes a cada opção (**c**) podem ser instruções compostas, sempre entre um **begin** e um **end.**

Veja este exemplo do uso de **case**:

```
program exemplo5;
var nodia:integer;
begin
 readln(nodia);
 if (nodia>0) and (nodia<8)
 then
  case nodia of
  1:writeln('Segunda-feira');
  2:writeln('Terça-feira');
  3:writeln('Quarta-feira');
  4:writeln('Quinta-feira');
  5:begin
   writeln('Sexta-feira');
   writeln('Dia de pagamento')
    end;
  6,7:begin
      writeln('Fim-de-semana');
      writeln('Descanso');
      end;
 end
 else begin
      writeln('Use valores');
      writeln('entre 1 e 7')
      end;
end.
```

Em algumas variações do Pascal, os rótulos 1, 2, 3, 4, 5, 6 e 7 devem ser colocados entre parênteses, fugindo, portanto, do Pascal padrão. Na verdade, em algumas implementações para a adaptação ao sistema BASIC, a mudança na sintaxe é necessária.

O programa **exemplo5** mostra como superar um problema bastante freqüente no uso do **case**. Muitas vezes, o identificador pode assumir valores não encontrados entre os rótulos de **case**, o que provocaria uma interrupção e uma mensagem de erro. Assim, utilizamos um desvio condicional **if...then...else** para mudar o curso do programa, caso os valores de **nodia** não se encontrem no intervalo apropriado. Observe também que foram empregados rótulos com mais de um valor e instruções compostas após alguns deles.

DESENVOLVIMENTO DE PROGRAMAS

Com um conhecimento mais detalhado dos procedimentos disponíveis no Pascal, podemos passar ao exame de um programa escrito nessa linguagem.

O programa que apresentamos a seguir verifica se uma palavra ou frase é um palíndromo — ou seja, se ela pode ser lida de trás para frente sem sofrer alteração (como "radar" ou "Socorrame, subi no ônibus em Marrocos"), ignorando-se os espaços em branco e os sinais de pontuação. Suponhamos que você queira testar se esta frase é um palíndromo:

Arara é ave rara

O algoritmo inicial poderia ser:

begin
leia uma cadeia de caracteres
teste se é um palíndromo
end

Essa frase inicial é bem simples, consistindo só de duas etapas, além do **begin** e do **end** requeridos pelo Pascal. Poderíamos elaborar um algoritmo mais detalhado para a primeira etapa:

início
leia o número de caracteres (n)
leia os caracteres colocando-os na matriz A(i)
fim

Traduzindo essa primeira etapa para o Pascal, temos:

```
readln(n)
for i:=1 to n do read(a[i]);
```

Passando para a segunda etapa, chegamos ao seguinte algoritmo:

início
faça crsc igual a 1
faça decr igual a n
enquanto crsc <decr faça
se a[crsc] é pontuação então
incremente crsc
senão
se a[decr] é pontuação então
decremente decr
senão
se a[crsc] = a[decr] então
incremente crsc
decremente decr
senão
faça a variável booleana (pal) igual a *false*
fim

Esse algoritmo poderia ser refinado para o Pascal desta maneira:

```
crsc:=1;
decr:=n;
while (crsc < decr) and pal do
if(a[crsc]=' ')or(a[crsc]='.')
or(a[crsc]=',')or(a[crsc]='"')
 then crsc:=crsc+1
 else
 if(a[decr]=' ')or(a[decr]='.')
 or(a[decr]=',')or(a[decr]='"')
  then decr:=decr-1
  else
   if a[crsc]=a[decr] then
   begin
   crsc:=crsc+1;
   decr:=decr-1;
   end
   else pal:=false
```

Precisaríamos de uma terceira etapa, na qual o programa mostraria o seguinte resultado:

se pal é igual a *true* então
escreva "palíndromo"
senão
escreva "não é palíndromo"

o que, em Pascal, corresponderia a:

```
if pal then
writeln('palindrome');
else
writeln('não há palíndrome')
```

O PROGRAMA COMPLETO

Refinando algumas partes anteriores e reunindo-as, chegamos ao seguinte programa em Pascal:

```
program exemplo6;
const comp=30;
var a:array [1..comp] of char;
    i,n,crsc,decr : integer;
    pal : boolean;
begin
pal:=true;
readln(n);
for i:=1 to n do read(a[i]);
crsc:=1;
decr:=n;
while (crsc < decr) and pal do
if(a[crsc]=' ')or(a[crsc]='.')
or(a[crsc]=',')or(a[crsc]='"')
 then crsc:=crsc+1
 else
 if(a[decr]=' ')or(a[decr]='.')
 or(a[decr]=',')or(a[decr]='"')
  then decr:=decr-1
  else
   if a[crsc]=a[decr] then
   begin
   crsc:=crsc+1;
   decr:=decr-1;
   end
   else pal:=false;
if pal then
writeln('palindrome')
else
writeln('nao ha palindrome')
end.
```

O Pascal apresenta um tipo de variável desconhecido do BASIC, a variável *booleana*, que não assume valores numéricos nem guarda caracteres. Ela assume o valor de resultados lógicos — *true* (verdadeiro) e *false* (falso). Note que é desnecessário dizer:

if pal = true then...

bastando apenas:

if pal then...

REFINANDO O PROGRAMA

Se o seu sistema Pascal dispõe de algumas formas adicionais de manipular dados, diversos aperfeiçoamentos podem ser incorporados ao programa. Entretanto, como está, ele deve funcionar na maioria dos sistemas.

A variável booleana é usada para desviar o programa do laço **while** quando se determina que **a[crsc]** não é igual a **a[decr].** Tudo o que precisa ser feito é tornar **pal** igual a *false* quando o teste **a[crsc] = a[decr]** apresentar esse resultado. No momento em que o programa voltar para o cabeçalho do laço, encontrará um desvio condicional dentro de **while**. A técnica empregada em BASIC seria a seguinte:

```
140...
150 FOR I=1 TO N
    ...
    ...
190 IF A(CRSC)<>A(DECR) THEN
 260
    ...
    ...
230 NEXT I
240 PRINT "PALINDROME"
250 GOTO 270
260 PRINT "NAO E PALINDROME"
270 END
```

Um algoritmo melhor estruturado poderia ser apresentado assim:

início
defina o conjunto pontuação
...
...
se a[crsc] estiver em pontuação então
incremente crsc
senão
se a[decr] estiver em pontuação então
decremente decr
senão
...

Uma declaração desse tipo economizaria exaustivas comparações do tipo:

if (a[crsc] = ' ') or (a[crsc] = ',') or (a[crsc] = '.') or (a[crsc] = '"') then...

O Pascal aceita a definição de conjuntos (alguns compiladores mais simples podem não aceitar essa declaração). Assim, modifique o programa anterior aproveitando essa definição:

```
program exemplo7;
const comp=30;
type carac = ' '..'z';
     simb = set of carac;
var a : array[1..comp] of char;
    i,n,crsc,decr : integer;
    pal : boolean;
    pont : simb;
begin
pont:=[' ',',','.','"'];
  pal:= true;
  readln(n);
  for i:=1 to n do read(a[i]);
  crsc:=1;
  decr:=n;
  while (crsc<decr) and pal do
   if a[crsc] in pont then
   crsc:=crsc+1
   else
    if a[decr] in pont then
```

```
    decr:=decr-1
    else
      if a[crsc]=a[decr] then
      begin
      crsc:=crsc+1;
      decr:=decr-1;
      end
      else pal:=false;
  if pal then
  writeln('palindrome')
  else
  writeln('nao ha palindrome')
end.
```

Finalmente, temos a melhor estrutura para o problema do palíndromo, originalmente dividido em três etapas:

*** entrar a cadeia de caracteres**
*** verificar se ela é um palíndromo**
*** mostrar o resultado**

Cada etapa foi refinada em etapas menores até que os resultados fossem as próprias declarações em Pascal e, juntas, formassem um programa. No nosso caso, o programa era bem simples. Porém, em situações mais complexas, é aconselhável construí-lo por partes semi-independentes (*procedures*) e testá-las separadamente. Um procedure é uma sub-rotina nomeada com um rótulo e declarada junto com as variáveis. Durante a execução do programa, basta chamar o procedure através de seu rótulo, como se fosse um comando. Essa técnica permite modularizar os programas, tornando-os bem estruturados.

Há dois grupos de sistemas Pascal. Um deles usa compiladores em código específico — ou seja, escritos exclusivamente para as máquinas em que irão rodar. Como várias máquinas são similares, usando sistemas operacionais CP/M ou MS DOS, as adaptações de uma para outra são muito simples.

O segundo grupo é conhecido como Pascal UCSD (Universidade da Califórnia, em San Diego). Nesse sistema, o Pascal é compilado para o código de um computador hipotético denominado *máquina p*, e seu código é conhecido por *código p*. Isso significa que o compilador é o mesmo para todas as máquinas. Na verdade, o UCSD é escrito em Pascal, possuindo um programa específico para cada máquina. Sua vantagem é a incrível compatibilidade. Porém, a velocidade da compilação nesse sistema é bem menor que a proporcionada pelo compilador específico.

EXPERIÊNCIAS COM O PASCAL

Se você possui um sistema Pascal para seu microcomputador, tente colocar em execução alguns dos exemplos anteriores. Para isso, depois de ter acessado o compilador Pascal, digite o programa e compile-o. Os programas devem funcionar para qualquer compilador que aceite letras minúsculas.

Os microcomputadores que usam o CP/M dispõem de um compilador poderoso, o TURBO PASCAL. Para acessá-lo, basta chamar do disquete o nome TURBO, seguido de <**ENTER**>. Você irá obter a seguinte saudação na tela:

TURBO Pascal system Version 2.00A
CP/M-80, Z80
Copyright (C) 1983, 1984 by
BORLAND Inc.
Include error messages (Y/N)?

Pode-se optar por mensagens de erro escritas por extenso (**Y**) ou simplesmente descritas pelos seus códigos (**N**). Se você estiver iniciando seu contato com o Pascal, digite **Y** e <**ENTER**>. O sistema apresentará um menu:

Logged drive:

Work file:
Main file:

Edit Compile Run Save
eXecute Dir Quit compiler Options

Text:
Free:

Para iniciar qualquer trabalho, você terá que digitar **W** seguido do nome de um arquivo. Se ele já existir no disquete, o computador irá trazê-lo para a memória; caso contrário, criará um arquivo com esse nome.

Suponhamos que você queira criar o arquivo **exemplo7**. Tendo escrito esse nome seguido de <**ENTER**>, entre no modo de edição digitando a letra **E**.

No manual de instruções que acompanha o compilador, você encontrará orientações para movimentar o cursor por meio das teclas. Recorra a ele para escrever seu programa. Ao terminar, saia do modo de edição usando o comando apropriado (veja manual). Antes de tentar compilar o programa, guarde-o no disquete digitando **S**.

Para compilar o programa, pressione a tecla **C**. Caso haja algum erro, o compilador indicará em que posição ele ocorreu. Depois de corrigi-lo, tente compilar o programa novamente. Se o computador indicar que completou a tarefa, execute o programa com **X**.

Lembre-se de que cada declaração precisa ser seguida de um ponto e vírgula, exceto quando depois dela houver um **end**. Uma instrução composta deve vir entre um **begin** e um **end**. Se você colocar um ponto e vírgula imediatamente após um **while**, um **repeat** ou um **for**, o computador entenderá que aquele é o fim da declaração e ignorará os comandos seguintes a serem repetidos.

ORGANIZAÇÃO DE PROJETOS

- CAMINHO CRÍTICO
- REDE PERT
- ATIVIDADES, TEMPOS E EVENTOS
- CÁLCULOS

Se você deseja montar e controlar qualquer tipo de projeto – da reforma de um automóvel à compra de um imóvel –, utilize este programa para se organizar e economizar tempo.

Algumas vezes temos que realizar tarefas que incluem várias ações, cada qual exigindo um tempo diferente de execução, e todas dependendo do sucesso das etapas anteriores. Sem uma boa dose de organização e planejamento, é quase impossível saber quais etapas executar, e em que ordem. A situação fica ainda mais difícil quando há um prazo para o término da atividade.

Se calcularmos o tempo de execução de todas as ações envolvidas, veremos que há sempre uma certa seqüência de eventos que determina o tempo total do projeto. A essa seqüência chamamos *caminho crítico*. Qualquer atraso ou adiantamento no caminho crítico se refletirá numa alteração do tempo tomado pelo projeto. Por outro lado, uma atividade que não pertença a essa seqüência especial poderá ser atrasada sem causar alterações no tempo total.

O cálculo do caminho crítico não é tão fácil se estivermos limitados ao papel e caneta, mas, com o auxílio do computador e do programa que aqui fornecemos, torna-se bem mais simples e rápido. O programa permite a construção de um banco de dados contendo todas as atividades, seu tempo de execução (ou estimativas, caso não se conheça o tempo real) e a ordem em que elas deverão ser executadas. Baseado nessas informações, calcula o caminho crítico bem como o tempo livre das atividades não-críticas (aquelas que não pertencem à seqüência especial). Esse último dado indica ao usuário por quanto tempo é possível interromper o projeto sem alterar seu prazo final.

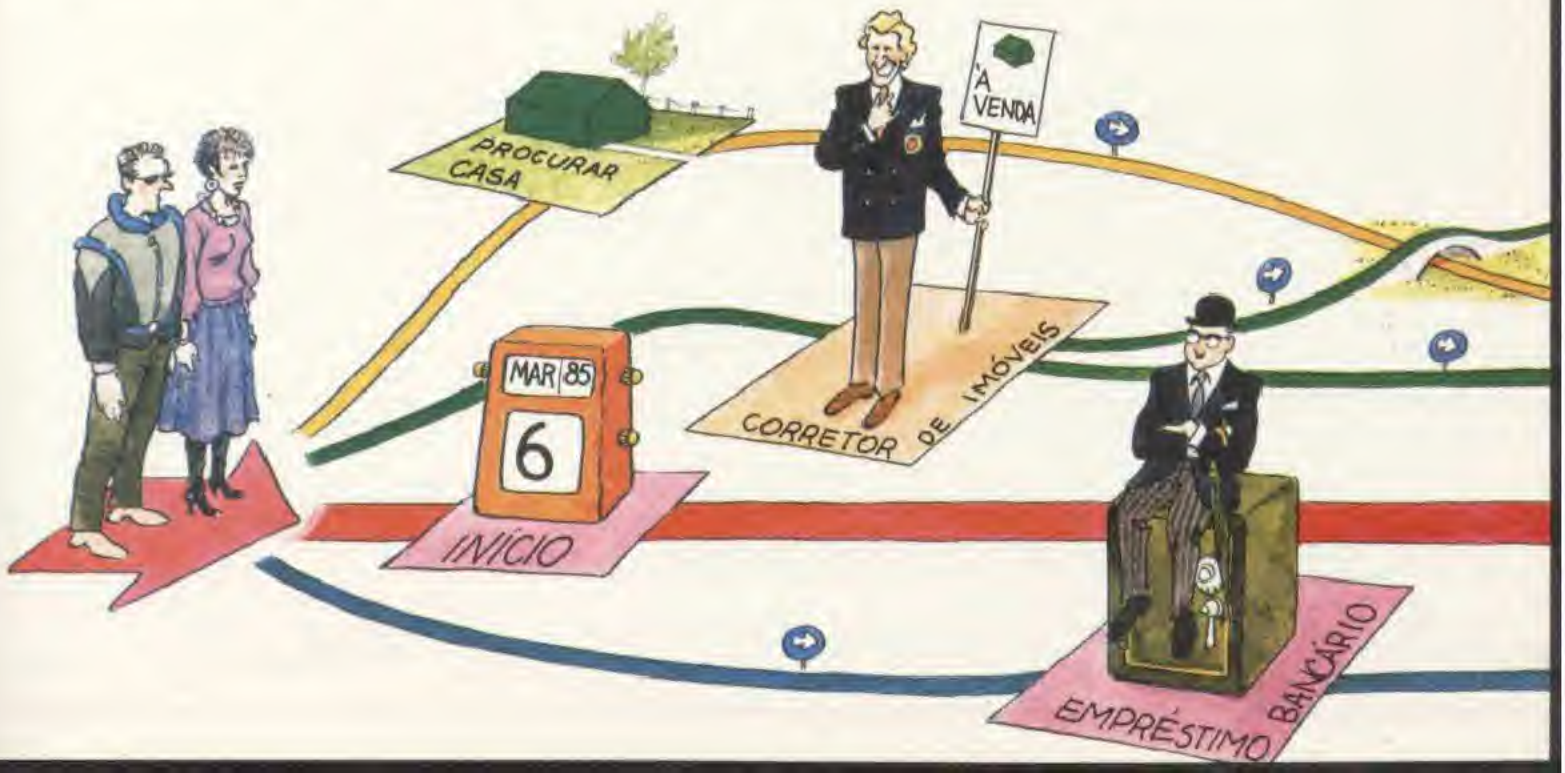
Para o cálculo do caminho crítico, utilizam-se as técnicas de programação conhecidas como CPM (do inglês *Critical Path Method*, Método do Caminho Crítico), CPA (do inglês *Critical Path Analysis*, Análise do Caminho Crítico) e PERT (do inglês *Program Evaluation and Review Technique*, Técnica de Inspeção e Estimativa de Projeto).

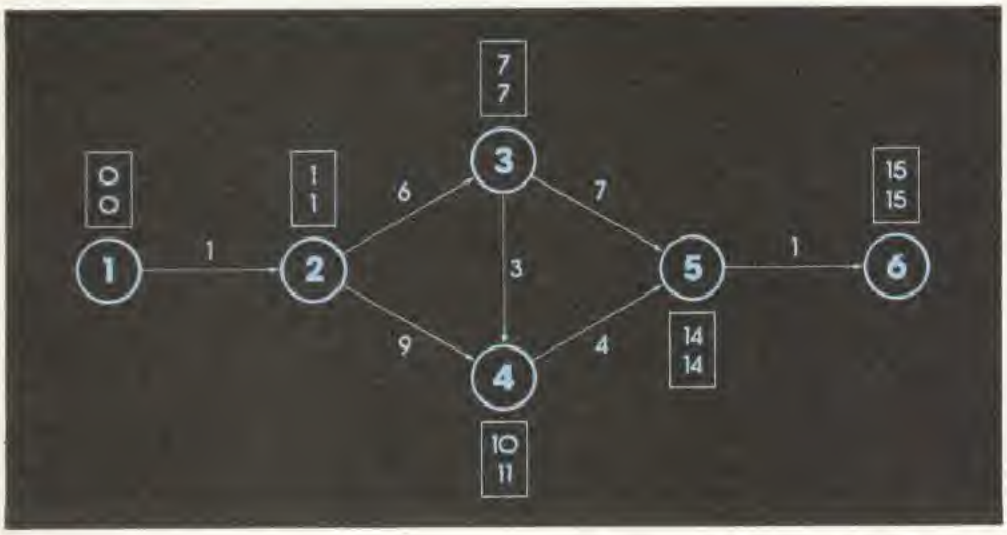
Não só projetos empresariais, mas qualquer projeto, por menor que seja, pode ser avaliado pelo programa.

Vamos tomar como exemplo um projeto que envolve a coordenação de várias etapas diferentes, muitas vezes sob prazos rigorosos: a compra de uma casa. O diagrama da página 1452 mostra como deveria ser o fluxo PERT para a execução dessa tarefa. Os círculos assinalam os *eventos*, ou seja, momentos entre atividades cujo tempo depende de outras. Eles marcam, assim, o início ou o fim de uma atividade. As atividades estão descritas ao longo das linhas que ligam os eventos, junto com as estimativas do tempo necessário.

Até aqui, muitas atividades parecem incertas e várias linhas se cruzam. Embora o diagrama contenha toda a informação necessária, ainda é difícil saber que atividades merecem prioridade. Não se pode garantir, também, que o projeto seja concluído dentro do prazo estipulado. Além disso, à medida que avançamos no projeto de aquisição da casa, é possível que surjam outros fatores capazes de afetar as expectativas iniciais, exigindo que o diagrama seja modificado e atualizado.

Nosso programa se encarrega de todas essas questões. Na primeira parte, apresentada a seguir, cuida do banco de dados das atividades; na segunda, calcula o caminho crítico.






```
 5 BORDER 0: PAPER 4: INK 0:
 CLS
 7 POKE 23658,8: POKE 23609,
 20
10 CLS : LET false=0: LET ma=
100: LET me=100: LET mh=212:
LET se=-1: LET fe=-1: GOSUB
12: LET ck=false: LET aa=0:
LET ee=0: GOTO 50
12 LET zz=9999: LET true=1:
LET p$="introduza ": LET a$="
 atividade"
14 DIM w$(85,32): LET w$(1)="
Nenhuma"+a$+"=PRECEDE o event
o": LET w$(2)="EXCEDEU"
16 LET w$(3)="VOCE NAO PODE U
SAR ESTE NUMERO ": LET w$(4)=
p$+"texto para este(a) "
18 LET w$(5)=a$+"REFERE A EVE
NTO NAO DEFINIDO "
22 DEF FN a(x)=x*(x<0):
DEF FN z(x)=x*(x>0)
26 DIM a(ma): DIM g(ma)
30 DIM w(ma): DEF FN w(x)=ABS
x*(x<1)+ABS (2-x)*(x>1):
 DEF FN x(x)=x*(2.37572+x*x*(
15.9402-x*x*(184.744-x*x*
688.472)))/1.20667
34 DEF FN I$(x)=(STR$ (x)+"
 ")( TO 6)
36 DEF FN b(x)=x-INT (x/256)*
256
38 DEF FN p$(x)="--"( TO INT
 ((6-x)/2)): DEF FN q$(x)="
"( TO INT ((6-x)/2))
40 DIM e(me)
42 DIM x(8)
44 DIM s(mh): DIM f(mh): DIM
u(mh): DIM t(mh): DIM n(mh):
DIM u$(mh,20): DIM y(mh): DIM
 z(mh)
46 DIM p(mh): DIM q(mh)
48 DEF FN u(x)=u(ABS x+(x=0))
*(x>0): RETURN
50 CLS : PRINT "1=define ";a$
'"2=apaga ";a$: PRINT "3=defi
ne evento"'"4=apaga evento"
60 PRINT "5=salva no gravador
"'"6=carrega do gravador"'"7=
testa gravador"'"8=mostra det
alhes"
62 PRINT "9=SAIDA"'"10=verifi
ca e ordena a rede"
64 PRINT "11=calc com tempos
medios"
66 PRINT "12=calc com incerte
zas": PRINT
70 INPUT t: PRINT t: IF t=9
THEN  STOP
72 IF t<1 OR t>12 THEN  PRINT
 "t=";t;" NAO RECONHECIDO":
GOTO 114
74 IF t>10 AND NOT ck THEN
PRINT "FACA VERIFICACAO DE DA
DOS PRIMEIRO": GOTO 114
76 IF aa=0 AND (t>7 OR t=5)
THEN  PRINT "IMPOSSIVEL - NAO
 FOI INTRODUZIDA NENHUMA";a$:
 GOTO 114
80 IF t>7 THEN  GOTO 100
82 IF t=6 THEN  CLEAR : LET t
=6
84 IF t>4 THEN  PRINT "Nome d
o arquivo:";: INPUT f$: PRINT
f$: GOTO 100
86 LET f$=a$: IF t>2 THEN
LET f$="evento"
88 PRINT p$;f$;" numero":
PRINT "ou zero para sair ";
90 INPUT u: PRINT u: LET u=
INT u: IF u=0 THEN  GOTO 50
92 IF u<1 OR u>zz THEN  PRINT
w$(3): GOTO 88
94 IF t>2 THEN  LET u=-u
96 GOSUB 450: LET ck=false
98 IF (t=2 OR t=4) AND (0=u(x
) OR zz<u(x)) THEN  PRINT "VO
CE NUNCA USOU ESTE NUMERO":
 GOTO 114
100 GOSUB 20*(t=1)+100*t+900*(
t-10)*(t>10)
112 GOTO 50
114 FOR t=1 TO 500: NEXT t:
GOTO 50
120 IF 0<u(x) AND zz>=u(x)
THEN  GOSUB 942: GOSUB 932:
GOTO 130
122 IF aa=ma THEN  PRINT w$(2)
;f$: RETURN
124 LET aa=a+1: LET a(aa)=x:
LET u(x)=u
130 PRINT w$(4);f$;":";: INPUT
u$(x): PRINT u$(x): LET xa=x
140 PRINT p$;"iniciar evento,
finalizar evento";: INPUT s,f:
PRINT s;" ";f: LET s=INT s:
LET f=INT f
142 IF s<1 OR s>zz OR f<1 OR f
>zz THEN  PRINT w$(3): GOTO
 140
150 LET u=-s: GOSUB 450: IF u(
x)<0 THEN  GOTO 156
152 IF ee=me THEN  PRINT w$(2)
;"eventos": GOTO 140
154 GOSUB 350
156 LET s(xa)=x
160 LET u=-f: GOSUB 450: IF u(
x)<0 THEN  GOTO 166
162 IF ee=me THEN  PRINT w$(2)
;"eventos": GOTO 140
164 GOSUB 350
166 LET f(xa)=x
170 PRINT p$;"tempo provavel p
ara executar ";: INPUT t(xa):
PRINT t(xa)
172 IF t(xa)<0 THEN  PRINT "NA
O PODE SER FEITO COM ESTA VELO
CIDADE": GOTO 170
180 PRINT "Introduza estimativ
a de tempo  com 90% de certez
a": PRINT "Pode ser executado
em ";: INPUT n(xa): PRINT n(xa
)
182 IF n(xa)<t(xa) THEN  PRINT
"ISTO NAO E COMPATIVEL COM O
   TEMPO PROVAVEL": GOTO 170
190 RETURN
200 FOR b=1 TO aa: IF x=a(b)
THEN  LET a=b
220 NEXT b: LET a(a)=a(aa):
LET u(x)=zz+1: LET aa=aa-1:
RETURN
300 IF u(x)<0 THEN  GOSUB 946:
GOSUB 933: GOTO 330
310 IF ee=me THEN  PRINT w$(2)
;f$: RETURN
312 GOSUB 350
```

```
5  CLEAR
10 MA = 100:ME = 100:FA = 0:MH
= 212: GOSUB 12:CK = FA: GOTO 5
0
12 TR = 1:Z$ =  CHR$ (13):DO$ =
 Z$ +  CHR$ (4):ZZ = 32766:QT$
=  CHR$ (34)
14 P$ = "FORNECA ":A$ = " ATIVI
DADE "
16  DIM W$(5):W$(1) = "NENHUMA"
 + A$ + " PRECEDE EVENTO":W$(2)
 = "EXCEDEU "
18 W$(3) = "NAO PODE USAR ESTE
NUMERO":W$(4) = P$ + "TEXTO PAR
A ESTE(A)"
20 W$(5) = " REFERE A EVENTO NA
O DEFINIDO "
22  DEF  FN A(X) = X * (X < 0):
 DEF  FN Z(X) = X * (X > 0)
28  DIM A(MA),G(MA)
30  DIM W(MA): DEF  FN W(X) =
```

```
ABS (X) * (X <  = 1) +  ABS (2
- X) * (X > 1)
32  DEF  FN X(X) = X * (2.37572
 + X * X * (15.9402 - X * X * (
184.744 - X * X * 688.472))) /
1.20667
40  DIM E(ME)
44  DIM S(MH),F(MH),U(MH),T(MH)
,N(MH),US(MH),Y(MH),Z(MH)
46  DIM P(MH),Q(MH)
48  DEF  FN U(X) = U( ABS (X) +
 (X = 0)) * (X > 0): RETURN
50  HOME : PRINT  TAB( 7);"MENU
 PRINCIPAL": PRINT : PRINT "01
= DEFINE";A$: PRINT "02 = DELET
A";A$: PRINT "03 = DEFINE EVENT
O": PRINT "04 = DELETA EVENTO"
52  PRINT "05 = SALVA DADOS": P
RINT "06 = CARREGA DADOS": PRIN
T "07 = DELETA ARQ DO DISCO": P
RINT "08 = MOSTRA DETALHES"
54  PRINT "09 = REINICIA"
56  PRINT "10 = VERIFICA E ORDE
NA REDE"
58  PRINT "11 = CALC COM DURACA
O MEDIA"
60  PRINT "12 = CALC COM INCERT
EZAS": PRINT "13 = SAIDA PARA "
;
62  IF KK$ = "S" THEN  INVERSE
: PRINT "IMPRESSORA": NORMAL
63  IF KK$ <  > "S" THEN  INVER
SE : PRINT "TELA": NORMAL
64  PRINT "14 = TERMINA"
66  PRINT : PRINT :T = 0: INPUT
 "SUA OPCAO (1-14) ";T
68  IF T = 14 THEN  INPUT "TEM
CERTEZA (S/N)? ";AN$: IF  LEFT$
 (AN$,1) = "S" THEN  HOME : END
69  IF T = 13 OR T = 14 THEN 10
0
72  IF T = 9 THEN 900
74  IF T < 1 OR T > 13 THEN  PR
INT "CODIGO";T;"NAO RECONHECIDO
": GOTO 114
76  IF T > 10 AND  NOT (CK) THE
N  PRINT "USE OPCAO (10) PRIMEI
RO": GOTO 114
78  IF AA = 0 AND (T > 7 OR T =
 5) THEN  PRINT "IMPOSSIVEL - N
ENHUMA";A$: GOTO 114
80  IF T > 7 THEN 100
82  IF T = 6 THEN  CLEAR :DO$ =
  CHR$ (13) +  CHR$ (4):T = 6
84  IF T > 4 THEN  HOME : PRINT
 "FORNECA NOME DO ARQ ";: INPUT
 F$: GOTO 100
86 F$ = A$: IF T > 2 THEN F$ =
" EVENTO"
88  HOME : PRINT P$;"NUMERO DO(
A)";F$: PRINT "OU ZERO PARA TER
MINAR"
90  INPUT U:U =  INT (U): IF U
= 0 THEN 50
92  IF U < 1 OR U > ZZ THEN  PR
INT W$(3): FOR E = 1 TO 1000: N
EXT : GOTO 88
94  IF T > 2 THEN U =  - U
96  GOSUB 450:CK = FA
98  IF (T = 2 OR T = 4) AND (0
= U(X) OR ZZ < U(X)) THEN  PRIN
T "VOCE NUNCA USOU ESTE NUMERO"
: GOTO 114
100  IF KK$ = "S" AND (T > 7 AN
D T < 12) THEN  PRINT DO$"PR#1"
101  HOME : ON T GOSUB 120,200,
300,400,500,600,700,800,900,100
0,2000,3000,960
105  IF KK$ = "S" AND (T > 7 AN
D T < 13) THEN  PRINT DO$"PR# 0
"
112  GOTO 50
114  FOR T = 1 TO 1000: NEXT :
GOTO 50
120  IF 0 < U(X) AND ZZ >  = U(
X) THEN  GOSUB 942: GOSUB 932:
GOTO 130
122  IF (AA = MA) THEN  PRINT W
$(2);F$: RETURN
124 AA = AA + 1:A(AA) = X:U(X)
= U
130  PRINT W$(4);F$: INPUT U$(X
):XA = X
140  PRINT P$;"EVENTO INICIAL ,
 EVENTO FINAL": INPUT S,F:S =
INT (S):F =  INT (F)
142  IF S < 1 OR S > ZZ OR F <
1 OR F > ZZ THEN  PRINT W$(3):
GOTO 140
150 U =  - S: GOSUB 450: IF U(X
) < 0 THEN 156
152  IF EE = ME THEN  PRINT W$(
2);"EVENTOS": GOTO 140
154  GOSUB 350
156 S(XA) = X
160 U =  - F: GOSUB 450: IF U(X
) < 0 THEN 166
162  IF EE = ME THEN  PRINT W$(
2);"EVENTOS": GOTO 140
164  GOSUB 350
166 F(XA) = X
170  PRINT P$;"TEMPO PROVAVEL D
E EXECUCAO": INPUT T(XA)
172  IF T(XA) < 0 THEN  PRINT "
MUITO RAPIDO !!": GOTO 170
180  PRINT "ESTIMATIVA COM 90%
DE CERTEZA": PRINT "PODE SER FE
ITO EM": INPUT N(XA)
182  IF N(XA) < T(XA) THEN  PRI
NT "NAO CONFERE COM O TEMPO PRO
VAVEL": GOTO 170
190  RETURN
200  FOR B = 1 TO AA: IF X = A(
B) THEN A = B
220  NEXT B:A(A) = A(AA):U(X) =
 ZZ + 1:AA = AA - 1: RETURN
300  IF U(X) < 0 THEN  PRINT "E
VENTOS":XP = U(X): GOSUB 950: P
RINT U$(X): GOTO 330
310  IF EE = ME THEN  PRINT W$(
2);F$: RETURN
312  GOSUB 350
330  PRINT W$(4);F$: INPUT U$(X
):S(X) = 0: RETURN
350 EE = EE + 1:E(EE) = X:S(X)
=  - 1:F(X) = 0:U(X) = U
360 T(X) = 0:N(X) = 0:U$(X) = "
": RETURN
400 Z = X: FOR F = 1 TO EE: IF
E(F) = Z THEN E = F
420  NEXT F:E(E) = E(EE):U(Z) =
 ZZ + 1:EE = EE - 1: RETURN
450 Z = U -  INT ((U - 1) / MH)
 * MH:Y = 2:X = 0
460  IF X = 0 AND (0 = U(Z) OR
ZZ + 1 = U(Z)) THEN X = Z
470  IF U = U(Z) THEN X = Z: RE
TURN
480  IF Y = 1 OR U(Z) = 0 THEN
 RETURN
490 Z = Z + Y - MH *  INT ((Z +
 Y - 1) / MH):Y = Y + Y - MH *
 INT ((Y + Y - 1) / MH): GOTO 4
60
500  HOME : PRINT DO$"OPEN "F$
505  PRINT DO$"DELETE "F$
510  PRINT DO$"OPEN "F$
515  PRINT DO$"WRITE "F$
520  PRINT MA;Z$;ME;Z$;MH;Z$;AA
;Z$;EE;Z$;CK
525  IF CK THEN  PRINT SE;Z$;FE
530  FOR A = 1 TO AA:X = A(A)
535  PRINT X;Z$U(X)Z$S(X)Z$F(X)
Z$T(X)Z$N(X)Z$G(A)Z$US(X)
540  NEXT A
545  FOR E = 1 TO EE:X = E(E)
550  PRINT X;Z$U(X)Z$S(X)Z$F(X)
Z$T(X)Z$N(X)Z$US(X)
555  NEXT E
560  FOR X = 1 TO MH: IF U(X)
 ZZ + 1 THEN  PRINT X
565  NEXT X: PRINT 0
570  PRINT DO$"CLOSE "F$
575  RETURN
```

```
4 MAXFILES=3
6 OPEN "CRT:" FOR OUTPUT AS#1
8 OPEN "LPT:" FOR OUTPUT AS#2
10 CLEAR 2000:MH=212:ME=100:MA=
100:FA=0:GOSUB 20:CK=FA:GOTO 14
0
20 ZZ=9999:TR=-1:P$="INTRODUZA
":A$=" ATIVIDADE":E$=CHR$(13)
30 DIMW$(5):W$(1)="NENHUMA"+A$+
" PRECEDE O EVENTO":W$(2)="EXCE
DEU"
40 W$(3)="VOCE NAO PODE USAR ES
TE NUMERO":W$(4)=P$+"TEXTO PARA
 ESTE(A) "
50 W$(5)="REFERE A EVENTO NAO D
EFINIDO"
60 DEFFNA(X)=-X*(X<0):DEFFNZ(X)
=-X*(X>0)
70 DIM A(MA),G(MA)
80 DIM W(MA):DEFFNW(X)=-ABS(X)*
(X<=1)-ABS(2-X)*(X>1)
90 DFFNX(X)=X*(2.37572+X*X*(15.
9402-X*X*(184.744-X*X*688.472))
)/1.20667
100 DIM E(ME)
110 DIM S(MH),F(MH),U(MH),T(MH)
,N(MH),US(MH),Y(MH),Z(MH)
120 DIM P(MH),Q(MH)
130 DEFFNU(X)=-U(ABS(X)-(X=0))*
(X>0):RETURN
140 CLS:PR=1:LOCATE 12,1:PRINT"
MENU PRINCIPAL":PRINT"1=DEFINE"
;A$;" OU EVENTO":PRINT"2=APAGA"
;A$;" OU EVENTO"
150 PRINT"3=SALVA DADOS":PRINT"
4=CARREGA DADOS":PRINT"5=IMPRIM
E DETALHES":PRINT"6=TERMINA"
160 PRINT"7=VERIFICA E ORDENA A
 REDE"
170 PRINT"8=CALC COM TEMPOS MED
```

```
IOS"
180 PRINT"9=CALC COM INCERTEZAS
"
190 TS=INKEY$:IF T$<"1" OR T$>"
9" THEN 190
200 T=VAL(T$):PRINT T
210 IF T>7 AND NOT(CK) THEN PRI
NT"FACA VERIFICACAO (7) PRIMEIR
O":GOTO 380
220 IF AA=0 AND (T>40 OR T=3) T
HEN PRINT"IMPOSSIVEL - NAO FOI
INTRODUZIDO";A$:GOTO380
230 IF T>4 THEN 350
240 IF T=4 THEN CLEAR 2000:T=4
250 IF T>2 THEN PRINT"NOME DO A
RQUIVO":INPUTF$:GOTO 350
260 CLS:PRINT A$;" OU EVENTO (A
/E) ?";
270 T$=INKEY$:IF T$<>"A" AND T$
<>"E" THEN 270
280 PRINT T$:IF T$="A" THEN F$=
A$ ELSE F$=" EVENTO"
290 PRINT:PRINT P$;F$;" NUMERO"
:PRINT"OU ZERO PARA SAIR"
300 INPUT U:U=INT(U):IF U=0 THE
N 140
310 IF U<1 OR U>ZZ THEN PRINT W
$(3):GOTO 280
320 IF T$="E" THEN U=-U
330 GOSUB 700:CK=FA
340 IF(T=2 OT T=4)AND(0=U(X) OR
 ZZ<U(X) THEN PRINT"VOCE NUNCA
USOU ESTE NUMERO":GOTO 380
350 ON T GOSUB 390,590,750,830,
890,960,1070,1550,1660
360 IF T<3 THEN 290
370 GOTO 140
380 FOR T=1 TO 1000:NEXT T:GOTO
 140
390 IF F$<>A$ THEN 620
400 IF 0<U(X) AND ZZ>=U(X) GOSU
B 1030:GOSUB 1000:GOTO 430
410 IF AA=MA THEN PRINTW$(2);F$
:RETURN
420 AA=AA+1:A(AA)=X:U(X)=U
430 PRINT W$(4);F$:INPUT U$(X):
XA=X
440 PRINT P$;"INICIAR EVENTO, F
INALIZAR EVENTO":INPUT S,F:S=IN
T(S):F=INT(F)
450 IF S<1 OR S>ZZ OR F<1 OR F>
ZZ THEN PRINT W$(3):GOTO 440
460 U=-S:GOSUB 700:IF U(X)<0 TH
EN 490
470 IF EE=ME THEN PRINT W$(2);"
EVENTOS":GOTO 440
480 GOSUB 660
490 S(XA)=X
500 U=-F:GOSUB 700:IF U(X)<0 TH
EN 530
510 IF EE=ME THEN PRINT W$(2);"
EVENTOS":GOTO 440
520 GOSUB 660
530 F(XA)=X
540 PRINT P$;"TEMPO PROVAVEL PA
RA EXECUTAR":INPUT T(XA)
550 IF T(XA)<0 THEN PRINT"NAO P
ODE SER FEITO COM ESTA VELO
CIDADE":GOTO 540
560 PRINT"INTRODUZA ESTIMATIVA
DE TEMPO  COM 90% DE CERTEZA":
PRINT"PODE SER FEITO EM":INPUT
N(XA)
570 IF N(XA)<T(XA) THEN PRINT "
NAO E COMPATIVEL COM O TEMPO
 PROVAVEL":GOTO 540
580 RETURN
```

```
10 PCLEAR 1:CLEAR 2000:MH=212:M
E=100:MA=100:FA=0:GOSUB 20:CK=F
A:GOTO 140
20 ZZ=9999:TR=-1:P$="INTRODUZA
":A$=" ATIVIDADE":E$=CHR$(13)
30 DIMW$(5):W$(1)="NENHUMA"+A$+
" PRECEDE O EVENTO":W$(2)="EXCE
DEU"
40 W$(3)="VOCE NAO PODE USAR ES
TE NUMERO":W$(4)=P$+"TEXTO PARA
 ESTE(A) "
50 W$(5)="REFERE A EVENTO NAO D
EFINIDO"
60 DEFFNA(X)=-X*(X<0):DEFFNZ(X)
=-X*(X>0)
70 DIM A(MA),G(MA)
80 DIM W(MA):DEFFNW(X)=-ABS(X)*
(X<=1)-ABS(2-X)*(X>1)
90 DFFNX(X)=X*(2.37572+X*X*(15.
9402-X*X*(184.744-X*X*688.472))
)/1.20667
100 DIM E(ME)
110 DIM S(MH),F(MH),U(MH),T(MH)
,N(MH),U$(MH),Y(MH),Z(MH)
120 DIM P(MH),Q(MH)
130 DEFFNU(X)=-U(ABS(X) (X=0))*
(X>0):RETURN
140 CLS:PR=0:PRINT @8,"MENU PRI
NCIPAL":PRINT"1=DEFINE";A$;" OU
 EVENTO":PRINT"2=APAGA";A$;" OU
 EVENTO"
150 PRINT"3=SALVA DADOS":PRINT"
4=CARREGA DADOS":PRINT"5=IMPRIM
E DETALHES":PRINT"6=SAIDA"
160 PRINT"7=VERIFICA E ORDENA A
 REDE"
170 PRINT"8=CALC COM TEMPOS MED
IOS"
180 PRINT"9=CALC COM INCERTEZAS
":PRINT"?";
190 T$=INKEY$:IF T$<"1" OR T$>"
9" THEN 190
200 T=VAL(T$):PRINT T
210 IF T>7 AND NOT(CK) THEN PRI
NT"FACA VERIFICACAO (7) PRIMEIR
O":GOTO 380
220 IF AA=0 AND (T>40 OR T=3) T
HEN PRINT"IMPOSSIVEL - NAO FOI
INTRODUZIDO";A$:GOTO380
230 IF T>4 THEN 350
240 IF T=4 THEN CLEAR 2000:T=4
250 IF T>2 THEN PRINT"NOME DO A
RQUIVO":INPUTF$:GOTO 350
260 CLS:PRINT A$;" OU EVENTO (A
/E) ?";
270 T$=INKEY$:IF T$<>"A" AND T$
<>"E" THEN 270
280 PRINT T$:IF T$="A" THEN F$=
A$ ELSE F$=" EVENTO"
290 PRINT:PRINT P$;F$;" NUMERO"
:PRINT"OU ZERO PARA SAIR"
300 INPUT U:U=INT(U):IF U=0 THE
N 140
310 IF U<1 OR U>ZZ THEN PRINT W
$(3):GOTO 280
320 IF T$="E" THEN U=-U
330 GOSUB 700:CK=FA
340 IF(T=2 OT T=4)AND(0=U(X) OR
 ZZ<U(X) THEN PRINT"VOCE NUNCA
USOU ESTE NUMERO":GOTO 380
350 ON T GOSUB 390,590,750,830,
890,960,1070,1550,1660
360 IF T<3 THEN 290
370 GOTO 140
380 FOR T=1 TO 1000:NEXT T:GOTO
 140
390 IF F$<>A$ THEN 620
400 IF 0<U(X) AND ZZ>=U(X) GOSU
B 1030:GOSUB 1000:GOTO 430
410 IF AA=MA THEN PRINTW$(2);F$
:RETURN
420 AA=AA+1:A(AA)=X:U(X)=U
430 PRINT W$(4);F$:INPUT U$(X):
XA=X
440 PRINT P$;"INICIAR EVENTO, F
INALIZAR EVENTO":INPUT S,F:S=IN
T(S):F=INT(F)
450 IF S<1 OR S>ZZ OR F<1 OR F>
ZZ THEN PRINT W$(3):GOTO 440
460 U=-S:GOSUB 700:IF U(X)<0 TH
EN 490
470 IF EE=ME THEN PRINT W$(2);"
EVENTOS":GOTO 440
480 GOSUB 660
490 S(XA)=X
500 U=-F:GOSUB 700:IF U(X)<0 TH
EN 530
510 IF EE=ME THEN PRINT W$(2);"
EVENTOS":GOTO 440
520 GOSUB 660
530 F(XA)=X
540 PRINT P$;"TEMPO PROVAVEL PA
RA EXECUTAR":INPUT T(XA)
550 IF T(XA)<0 THEN PRINT"NAO P
ODE SER FEITO COM ESTA VELO
CIDADE":GOTO 540
560 PRINT"INTRODUZA ESTIMATIVA
DE TEMPO  COM 90% DE CERTEZA":
PRINT"PODE SER FEITO EM":INPUT
N(XA)
570 IF N(XA)<T(XA) THEN PRINT "
NAO E COMPATIVEL COM O TEMPO
 PROVAVEL":GOTO 540
580 RETURN
```

PLANEJAMENTO DA REDE PERT

Antes de usar o programa para avaliar um projeto, você terá que dividi-lo em atividades individuais e estimar o tempo de execução de cada uma delas. Para isso, o mais conveniente, como já vimos, é desenhar um esquema de blocos: a rede PERT.

À medida que for desenhando o diagrama, descreva as atividades ao longo das linhas de conexão entre eventos. Dentro dos círculos, coloque a descrição dos eventos. Se estes forem apenas passos intermediários entre atividades, não é necessário atribuir-lhes nomes específicos.

A versão do programa para o Spectrum permite o uso de até vinte caracte-

res por evento ou atividade; nos demais micros, podem-se utilizar 255 caracteres. Entretanto, dê preferência a títulos sucintos, para não sobrecarregar a memória disponível.

Se você tem uma noção mais ou menos precisa da duração de cada atividade, escreva esse valor no diagrama. Posteriormente, veremos como estimar os tempos. Lembre-se, porém, de que a unidade de medida de tempo escolhida (hora, dia, semana etc.) deve ser sempre a mesma em toda a rede.

O programa só funcionará corretamente se a rede for viável do ponto de vista lógico — deve ter apenas um ponto de partida e um de chegada, e não formar laços ou caminhos circulares.

O passo seguinte ao desenho da rede PERT é a numeração de todas as atividades e eventos e sua introdução no programa. A ordem dos números não é importante, mas o computador precisa deles para trabalhar. Um método bastante simples consiste em numerar os eventos de dez em dez, como se faz com um programa em BASIC — o que permite inserir números extras entre os já existentes, se for necessário.

Quando você for definir os eventos e as atividades, o computador pedirá o número e a descrição de cada um deles, além de uma estimativa do seu tempo médio e do seu tempo pessimista (aquele que conta com 90% de possibilidade de não ser ultrapassado).

Na vida real, raramente podemos calcular com precisão o tempo que levaremos para realizar uma atividade — mesmo que a tenhamos executado muitas vezes. Entretanto, não é difícil estimar sua duração média, bem como sua duração aproximada. Isso é tudo o que o programa exige em termos de estimativa de tempo. No entanto, como você verá, os cálculos que ele faz não são tão simples, pois devem levar em conta quatro diferentes situações.

A primeira situação é aquela em que estamos absolutamente seguros quanto ao tempo que uma certa atividade levará. Por exemplo, se as instruções para a construção de uma casa recomendam que deixemos o cimento secar por 48 horas, é exatamente isso que vamos fazer! Nesse caso, ao usar o programa, devemos indicar o mesmo valor para o tempo médio e o pessimista.

A segunda situação ocorre quando estamos razoavelmente seguros sobre a duração de determinada atividade. Sabemos, por exemplo, que uma viagem entre as cidades de São Paulo e Campinas, de automóvel, demora cerca de 90 minutos, porque já a repetimos muitas vezes, sem observar grandes variações (mais ou menos 10%, devido a eventuais dificuldades de trânsito). Portanto, entramos 90 minutos como tempo médio, e 100 minutos como tempo pessimista. Essa situação corresponde à curva conhecida como curva *normal* ou *gaussiana*.

A terceira situação é a que chamamos "espere até acontecer". Por exemplo: você só terá a certeza de que a reforma no telhado foi bem feita quando começar a chover. Aqui, o tempo pessimista é cerca de 2,5 vezes maior que o tempo médio. Este pode ser obtido a partir do número de dias de chuva naquele mês específico.

A quarta e última situação corresponde ao tempo do "tudo ou nada". É muito improvável, por exemplo, que uma determinada peça do carro quebre, mas, se isso ocorrer, o veículo ficará na oficina durante dez dias. Nesse caso, tome o valor máximo (dez dias) como o tempo pessimista, e a média aritmética (10 vezes 1/100, ou 1/10 de dia) como o tempo médio.

Não é necessário indicar ao microcomputador o gráfico em que se encaixa cada atividade. A máquina simplesmente toma as duas estimativas de tempo, e efetua os cálculos. Se a diferença entre os dois valores for de 0 a 30%, ele usa a curva gaussiana; se for de 30 a 130%, usa uma curva gaussiana modificada. Por outro lado, se o valor da média estiver entre 130 e 300%, o programa recorre à curva exponencial e, se estiver acima dessa porcentagem, utiliza a curva bimodal.

Depois de entrar todas as atividades e suas respectivas estimativas de tempo, você deverá entrar os eventos. Para isso, basta digitar o número e a descrição de cada evento no diagrama. Se você cometer algum erro, chame as opções de apagamento ou de modificação de eventos ou atividades.

A informação entrada no computador pode ser exibida em vários tipos de tabela. A opção "Mostre Detalhes" mostrará uma lista de tudo o que foi entrado. Se você tem uma impressora, poderá também obter uma cópia em papel.

CONSISTÊNCIA DE DADOS

Antes que o computador prossiga com os cálculos, é necessário checar se há consistência lógica na rede entrada. Se esta contiver caminhos circulares, o programa cairá em uma alça sem fim de cálculos e, eventualmente, apresentará um defeito de execução.

Escolha a opção de verificação e observe o resultado. Se tudo estiver bem, o programa imprimirá os números dos eventos de início e de fim. Entretanto, se encontrar alguma inconsistência na rede, emitirá uma mensagem de erro, identificando caminhos circulares ou interrupções.

O CAMINHO CRÍTICO

Finalmente, selecione a opção para calcular o caminho crítico. Existem duas alternativas: a primeira usa o tempo médio de cada atividade, e a segunda, o tempo incerto (pessimista). Peça antes a primeira alternativa.

O vídeo mostrará todas as atividades, seu número de código e descrição. Em seguida, dirá quando cada uma deve ser iniciada e concluída, se existe alguma folga, e se a atividade é crítica — ou seja, se está no caminho crítico, com folga igual a zero.

O tempo é expresso na mesma unidade que foi entrado. Assim, se você usou dias, todos os tempos irão se referir a dias, contados a partir do evento inicial da rede. Suponhamos que a tabela indique que a atividade 3 deve começar no dia 6, terminar no dia 10, e tem uma folga de dois dias. Você poderá começar aquela atividade no mínimo seis dias depois de iniciado o projeto, com uma tolerância de dois dias (ou seja, se você atrasar a atividade até o dia 8, o tempo total do projeto não será alterado).

As atividades que têm folga zero devem começar exatamente no dia indicado; caso contrário, haverá um atraso no projeto. Se isso ocorrer, você precisará recalcular toda a rede, com a nova estimativa para a atividade ou atividades infratoras.

Se a maioria dos tempos que você entrou são incertos, e o tempo pessimista é muito distinto do médio, use a segunda opção de cálculo, pois o caminho crítico será diferente.

Quando você escolhe essa opção, o computador considera cada atividade e, usando o gráfico de distribuição mais apropriado, seleciona um tempo ao acaso, dentro dos limites propostos. A partir desse tempo, ele calcula o caminho crítico para toda a rede, exatamente como na opção anterior, e armazena os valores de início e de folga das atividades. O processo se repete 44 vezes: em cada uma delas um novo número aleatório é escolhido para as atividades incertas. As 45 repetições (todas são exibidas na tela) proporcionam uma amostra razoável para a derivação dos valores médios de tempo.

Os tempos de início e fim do projeto correspondem à média de 45 repetições

ao acaso; portanto, são bastante confiáveis. A folga média das atividades não críticas também é indicada.

Os valores críticos mostram a porcentagem da participação de certa atividade no caminho crítico — 100% significa que a atividade será sempre crítica; 0%, que ela nunca é crítica.

O último valor mostra o desvio padrão do tempo de folga, indicando a variação possível da folga e, também, a confiabilidade da estimativa. Por exemplo, se há uma folga de 1,5 dia e o desvio é igual a 1, a folga pode variar entre 0,5 dia e 2,5 dias. Nesse caso, conclui-se que a folga é pouco confiável. Ao contrário, um desvio padrão de 0,1 mostra que o valor atribuído à folga é bastante confiável.

```
330 PRINT w$(4);f$;":";: INPUT
 u$(x): PRINT u$(x): LET s(x)=
0: RETURN
350 LET ee=ee+1: LET e(ee)=x:
LET s(x)=-1: LET f(x)=0: LET u
(x)=u
360 LET t(x)=0: LET n(x)=0:
LET u$(x)="": RETURN
400 LET z=x: FOR f=1 TO ee: IF
e(f)=z THEN  LET e=f
420 NEXT f: LET e(e)=e(ee):
LET u(z)=zz+1: LET ee=ee-1:
RETURN
450 LET z=u-INT ((u-1)/mh)*mh:
LET y=2: LET x=0
460 IF x=0 THEN  IF 0=u(z) OR
zz+1=u(z) THEN  LET x=z
470 IF u=u(z) THEN  LET x=z:
RETURN
480 IF y=1 OR 0=u(x) THEN
RETURN
490 LET z=z+y-mh*INT ((z+y-1)/
mh): LET y=y+y-mh*INT ((y+y-1)
/mh): GOTO 460
500 LET x(1)=ma: LET x(2)=me:
LET x(3)=mh: LET x(4)=aa: LET
x(5)=ee: LET x(6)=ck: LET x(7)
=se: LET x(8)=fe: PRINT "press
ione <ENTER> dez vezes": SAVE
f$+"x" DATA x(): FOR x=1 TO
100: NEXT x
510 SAVE f$+"a" DATA a(): SAVE
f$+"e" DATA e(): SAVE f$+"f"
DATA f(): SAVE f$+"g" DATA g()
: SAVE f$+"n" DATA n(): SAVE
f$+"s" DATA s(): SAVE f$+"t"
DATA t(): SAVE f$+"u" DATA u()
: SAVE f$+"u$" DATA u$():
 RETURN
600 LOAD f$+"x" DATA x(): LET
ma=x(1): LET me=x(2): LET mh=x
(3): LET aa=x(4): LET ee=x(5):
LET ck=x(6): LET se=x(7): LET
fe=x(8): GOSUB 12
610 LOAD f$+"a" DATA a(): LOAD
f$+"e" DATA e(): LOAD f$+"f"
DATA f(): LOAD f$+"g" DATA g()
: LOAD f$+"n" DATA n(): LOAD
f$+"s" DATA s(): LOAD f$+"t"
DATA t(): LOAD f$+"u" DATA u()
: LOAD f$+"u$" DATA u$(): LET
false=0: RETURN
700 LET x(1)=ma: LET x(2)=me:
LET x(3)=mh: LET x(4)=aa: LET
x(5)=ee: LET x(6)=ck: LET x(7)
=se: LET x(8)=fe: VERIFY f$+"x
" DATA x()
710 VERIFY f$+"a" DATA a():
VERIFY f$+"e" DATA e(): VERIFY
f$+"f" DATA f(): VERIFY f$+"g"
DATA g(): VERIFY f$+"n" DATA n
(): VERIFY f$+"s" DATA s():
VERIFY f$+"t" DATA t(): VERIFY
f$+"u" DATA u(): VERIFY f$+"u$
" DATA u$(): RETURN
800 GOSUB 942
810 FOR a=1 TO aa: LET x=a(a):
GOSUB 932
820 LET y=y+1+(LEN u$(x)>4):
IF y>20 AND a<aa THEN  GOSUB
 940: GOSUB 942
830 NEXT a: GOSUB 940
840 GOSUB 946: FOR e=1 TO ee:
LET x=e(e): GOSUB 933
850 LET y=y+1+(LEN u$(x)>4):
IF y>20 AND e<ee THEN  GOSUB
940: GOSUB 946
860 NEXT e: GOTO 940
932 PRINT FN I$(FN u(s(x)));FN
I$(FN u(f(x)));FN I$(t(x));FN
I$(n(x));ABS u(x);" ";u$(x):
RETURN
933 PRINT ABS u(x),u$(x):
RETURN
940 PRINT "pressione <ENTER> p
ara sair": INPUT f$: CLS :
RETURN
942 CLS : PRINT "INICI FINAL T
EMPO 90% CODIGO TEXTO"
944 PRINT "FINAL INICI PROVL E
STIM ": LET y=3: RETURN
946 PRINT "CODIGO","TEXTO":
RETURN
 948 PRINT "PREV FINAL MIN  MAX
": LET y=3: RETURN
1000 LET ck=true: FOR a=1 TO aa
: LET x=a(a)
1020 LET z=s(x): IF s(z)<0 OR z
z<u(z) THEN  PRINT u(x);w$(5);u
(z): LET ck=false
1030 LET z=f(x): IF s(z)<0 OR z
z<u(z) THEN  PRINT u(x);w$(5);u
(z): LET ck=false
1040 NEXT a: IF ck=false THEN
GOTO 1750
1050 LET e=1
1060 LET z=e(e): IF s(z)<0 THEN
  GOSUB 400: IF e<=ee THEN  GOT
O 1060
1070 LET e=e+1: IF e<=ee THEN
GOTO 1060
1080 FOR e=1 TO ee: LET z=e(e):
 LET s(z)=0: LET f(z)=0: NEXT e
1082 FOR a=1 TO aa: LET x=a(a):
 LET s(f(x))=x: NEXT a
1090 LET se=0: FOR e=1 TO ee: L
ET z=e(e): IF s(z)>0 THEN  GOTO
 1096
1092 IF se=0 THEN  LET se=z: GO
TO 1096
1094 PRINT w$(1);u(z): IF se<=m
h THEN  PRINT w$(1);u(se): LET
se=mh+1
1096 NEXT e: IF se=0 THEN  PRIN
T "TODOS OS EVENTOS TEM";a$;" P
RECEDENDO"
1098 IF se=0 OR se>mh THEN  GOT
O 1750
1100 FOR e=1 TO ee: LET z=e(e):
 LET t(z)=0: LET n(z)=0: NEXT e
: LET t(se)=1
1110 LET last=1: FOR c=2 TO ee+
2: IF last<>c-1 THEN  GOTO 1170
1120 FOR a=1 TO aa: LET x=a(a):
 LET y=s(x): IF t(y)<>c-1 THEN
 GOTO 1160
1130 IF y=f(x) THEN  GOSUB 1200
: GOTO 1160
1140 IF y<>se THEN  LET y=s(y):
 GOTO 1130
1150 LET y=f(x): LET s(y)=s(x):
 LET f(s(y))=y: LET t(y)=c: LET
 fe=y: LET last=c
1160 NEXT a
1170 NEXT c: PRINT "evento inic
ial=";u(se);"evento final=";u(f
e)
1180 FOR e=1 TO ee: LET y=e(e)
1190 IF f(y)=0 AND y<>fe THEN
PRINT u(y);"NAO CONECTADO AO EV
ENTO FINAL": LET ck=false
1192 NEXT e: IF ck THEN  GOTO 1
300
1194 GOTO 1750
1200 CLS : PRINT "EXISTE O SEGU
INTE LOOP": PRINT "EVENTOS...":
 LET xa=a(a)
1210 LET x=f(xa): PRINT u(x): L
ET y=s(xa): PRINT u(y)
1220 LET y=s(y): PRINT u(y): IF
 y<>x THEN  GOTO 1220
1230 RETURN
1300 LET k=1: LET ak=aa: IF aa=
1 THEN  LET k=0
1310 LET ak=INT ((ak+k)/2): IF
ak=0 THEN  GOTO 1500
1320 LET k=0: FOR a=ak+1 TO aa:
 LET b=a-ak: LET x=a(a): LET y=
a(b): LET xe=s(x): LET ye=s(y)
1330 IF t(ye)+ye/zz<=t(xe)+xe/z
z THEN  GOTO 1360
1340 LET a(a)=y: LET a(b)=x: LE
T k=1
1360 NEXT a: GOTO 1310
1500 LET n(fe)=last: FOR d=last
-1 TO 1 STEP -1
1520 FOR a=1 TO aa: LET x=a(a):
 IF n(f(x))<>d+1 THEN  GOTO 156
0
1550 LET y=s(x): LET t(y)=t(x):
 LET n(y)=d
1560 NEXT a: NEXT d
1600 FOR a=1 TO aa: LET g(a)=a(
a): NEXT a: LET k=1: LET ak=aa:
 IF aa=1 THEN  LET k=0
1610 LET ak=INT ((ak+k)/2): IF
ak=0 THEN  GOTO 1700
1620 LET k=0: FOR a=ak+1 TO aa:
 LET b=b-ak: LET x=g(a): LET y=
g(b): LET xe=f(x): LET ye=f(y)
1630 IF n(ye)+ye/zz<=n(xe)+xe/z
z THEN  GOTO 1660
1640 LET g(a)=y: LET g(b)=x: LE
```

```
T k=1
1660 NEXT a: GOTO 1610
1700 LET ck=true: RETURN
1750 LET ck=false: PRINT AT 21,
8;"QUALQUER TECLA PARA CONTINUA
R": PAUSE 0: RETURN
2000 FOR a=1 TO aa: LET x=a(a):
 LET z(x)=t(x): NEXT a: GOSUB 2
100
2020 FOR a=1 TO aa: LET x=a(a):
 LET y(x)=(z(f(x))-y(s(x))=z(x)
)*100: NEXT a
2030 FOR b=1 TO aa STEP 5: CLS
: FOR a=b TO aa+FN a(b+4-aa): L
ET x=a(a)
2040 PRINT a$;" ";u(x);"=";u$(x
)( TO 16)
2050 LET c=y(s(x)): LET d=z(f(x
)): PRINT "pode iniciar ";c;",d
eve terminar ";d
2060 PRINT "tempo livre ";d-c-z
(x);" (critico ";y(x);"%)": IF
t=12 THEN  PRINT "desvio=";q(x)
;
2070 PRINT : NEXT a: GOSUB 940:
 NEXT b: RETURN
2100 FOR e=1 TO ee: LET y(e(e))
=0: NEXT e
2110 FOR a=1 TO aa: LET x=a(a):
 LET y(f(x))=y(f(x))+FN z(y(s(x
))-y(f(x))+z(x)): NEXT a
2120 FOR e=1 TO ee: LET z(e(e))
=y(e): NEXT e: FOR a=aa TO 1 S
TEP -1: LET x=q(a)
2130 LET z(s(x))=z(s(x))+FN a(z
(f(x))-z(s(x))-z(x)): NEXT a: R
ETURN
3000 FOR a=1 TO aa: LET x=a(a):
 LET p(x)=0: LET q(x)=0: LET y(
x)=0: NEXT a
3020 FOR e=1 TO ee: LET z=e(e):
 LET p(z)=0: LET q(z)=0: NEXT e
3030 FOR m=1 TO 43 STEP 3: FOR
a=1 TO aa: LET w(a)=2*RND-1: NE
XT a
3040 FOR n=0 TO 4 STEP 2: CLS :
 PRINT "CASO # ";m+n/2;" EM 45"
3050 FOR a=1 TO aa: LET x=a(a):
 LET tx=t(x): IF tx=0 THEN  LET
 z(x)=0: GOTO 3080
3052 LET nx=n(x): IF nx=tx THEN
  LET z(x)=tx: GOTO 3080
3054 LET w=FN w(w(a)+n/3): IF n
x>=tx*3 THEN  LET z(x)=n(x)*(w<
tx/nx): GOTO 3080
3060 IF nx>tx*2.34 THEN  LET z(
x)=-1x*LN w: GOTO 3080
3070 LET w=FN x(w-.5): LET z(x)
=ABS (tx+w*(nx-tx))
3080 NEXT a
3090 GOSUB 2100
3100 FOR a=1 TO aa: LET x=a(a):
 LET z=z(f(x))-y(s(x))-z(s)
3110 LET p(x)=p(x)+z: LET q(x)=
q(x)+z*z: LET y(x)=y(x)+(z<1.e-
6): NEXT a
3120 FOR e=1 TO ee: LET z=e(e):
 LET p(z)=p(z)+y(z): LET q(z)=q
(z)+z(z): NEXT e: NEXT n: NEXT
m
3200 FOR e=1 TO ee: LET z=e(e):
 LET y(z)=VAL (FN I$(p(z)/45))
3210 LET z(z)=VAL (FN I$(q(z)/4
5)): NEXT e
3220 FOR a=1 TO aa: LET x=a(a):
 LET y=y(x): LET y(x)=VAL ((STR
$ (y/45*100)+"    ")( TO 4))
3230 IF p(x)<1.e-2 THEN  LET p(
x)=0
3240 LET z=(45-y)+1.e-9: LET z(
x)=z(f(x))-y(s(x))-VAL (FN I$(p
(x)/Z))
3250 LET q(x)=SQR ABS ((q(x)-p(
x)*p(x)/z)/((z-1)+.1e-9)): IF q
(x)<1.e-6 THEN  LET q(x)=0
3260 NEXT a: GOTO 2030
```

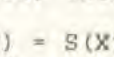
```
590 IF F$<>A$ THEN 680
600 FOR B=1 TO AA:IF X=A(B) THE
N A=B
610 NEXT B:A(A)=A(AA):U(X)=ZZ+1
:AA=AA-1:RETURN
620 IF U(X)<0 GOSUB 1050:PRINT
USING(#####  ";ABS(U(X));:PRINT
 U$(X):GOTO 650
630 IF EE=ME THEN PRINT W$(2);F
$:RETURN
640 GOSUB 660
650 PRINT W$(4);F$:INPUT U$(X):
S(X)=0:RETURN
660 EE=EE+1:E(EE)=X:S(X)=-1:F(X
)=0:U(X)=U
670 T(X)=0:N(X)=0:U$(X)="":RETU
RN
680 Z=X:FOR F=1 TO EE:IF E(F)=Z
 THEN E=F
690 NEXT F:E(E)=E(EE):U(Z)=ZZ+1
:EE=EE-1:RETURN
700 Z=U-INT((U-1)/MH)*MH:Y=2:X=
0
710 IF X=0 (0=U(Z) OR ZZ+1=U(Z)
)THEN X=Z
720 IF U=U(Z) THEN X=Z:RETURN
730 IF Y=1 OR 0=U(Z) THEN RETUR
N
740 Z=Z+Y-MH*INT((Z+Y-1)/MH):Y=
Y+Y-MH*INT((Y+Y-1)/MH):GOTO 170
750 OPEN "O",#-1,F$:PRINT #-1,M
A;ME;MH;AA;EE;CK
760 IF CK THEN PRINT#-1,SE;FE
770 FOR A=1 TO AA:X=A(A):PRINT
#-1,X;U(X);S(X);F(X);T(X);N(X);
G(A);U$(X):NEXT A
780 FOR E=1 TO EE:Z=E(E):PRINT#
-1,Z;U(Z);S(Z);F(Z);T(Z);N(Z);U
$(Z):NEXT E
790 FOR X=1 TO MH:IF U(X)=ZZ+1
THEN PRINT #-1,X
800 BEXT X:PRINT#-1,0
810 CLOSE#-1:MOTORON:FOR X=1 TO
 100:NEXT X:MOTOROFF:RETURN
820 CLS:PRINT"SALVAR DADOS - er
ro:FOR K=1 TO 1000:NEXT:RETURN
830 OPEN"I",#-1,F$:INPUT#-1,MA,
ME,MH,AA,EE,CK:GOSUB 20
840 IF CK THEN INPUT #-1,SE,FE
850 FOR A=1 TO AA:INPUT #-1,X,U
(X),S(X),F(X),T(X),N(X),G(A),US
(X):A(A)=X:NEXT A
860 FOR E=1 TO EE:INPUT#-1,Z,U(
Z),S(Z),F(Z),T(Z),N(Z),U$(Z):E(
E)=Z:NEXT E
870 INPUT #-1,X:IF X>0 THEN U(X
)=ZZ+1:GOTO 870
880 CLOSE #-1:RETURN
890 GOSUB 1870:A=0:GOSUB 1030
900 FOR A=1 TO AA:X=A(A):GOSUB
1000
910 Y=Y+1:IF Y>8 AND A<AA GOSUB
 1020:GOSUB 1030
920 NEXT A:GOSUB 1020
930 E=0:GOSUB 1050:FOR E=1 TO E
E:X=E(E):PRINT #PR,USING"#####
  ";ABS(U(X));:PRINT #PR,U$(X)
940 Y=Y+1:IF Y>15 AND E<EE GOSU
B 1020:GOSUB 1050
950 NEXT E:GOTO 1020
960 CLS:PRINT"TEM CERTEZA ?  (S
/N)"
970 T$=INKEY$:IF T$<>"S" AND T$
<>"N" THEN 970
980 IF T$="N" THEN RETURN
990 CLS:END
1000 PRINT #PR,USING"##### #### 
# ##### #####  #####";FNU(S(X))
,FNU(F(X)),T(X),N(X),ABS(U(X));
:IF PR=0 THEN PRINT
1010 PRINT #PR," ";U$(X):RETURN
1020 IF PR=0 THEN PRINT"<ENTER>
 PARA CONTINUAR":INPUT F$:CLS:R
ETURN ELSE RETURN
1030 CLS:IF PR=0 OR A=0 THEN PR
INT #PR,"INICIO ULTIMA VEZ   90
% ATIVIDADE":Y=3
1040 RETURN
1050 CLS:IF PR=0 OR E=0 THEN PR
INT #PR,"  EVENTO  TEXTO":Y=3
1060 RETURN
1070 CK=TR:FOR A=1 TO AA:X=A(A)
1080 Z=S(X):IF S(Z)<0 OR ZZ<U(Z
) THEN PRINT A$;U(X);W$(5);U(Z)
:CK=FA
1090 Z=F(X):IF S(Z)<0 OR ZZ<U(Z
) THEN PRINT A$;U(X);W$(5);U(Z)
:CK=FA
1100 NEXT A:IF CK=FA THEN 1540
1110 E=1
1120 Z=E(E):IF S(Z)<0 GOSUB 680
:IF E<=EE THEN 1120
1130 E=E+1:IF E<=EE THEN 1120
1140 FOR E=1 TO EE:Z=E(E):S(Z)=
0:F(Z)=0:NEXT E
1150 FOR A=1 TO AA:X=A(A):S(F(X
))=X:NEXT A
1160 SE=0:FOR E=1 TO EE:Z=E(E):
IF S(Z)>0 THEN 1190
1170 IF SE=0 THEN SE=Z:GOTO 119
0
1180 PRINT W$(1);U(Z):IF SE<=MH
 THEN PRINT W$(1);U(SE):SE=MH+1
1190 NEXT E:IF SE=0 THEN PRINT"
TODOS OS EVENTOS TEM":PRINT A$;
"PRECEDENDO"
1200 IF SE=0 OR SE>MH THEN 1540
1210 FOR E=1 TO EE:Z=E(E):T(Z)=
0:N(Z)=0:NEXT E:T(SE)=1
1220 LA=1:FOR C=2 TO EE+2:IF LA
<>C-1 THEN 1280
1230 FOR A=1 TO AA:X=A(A):Y=S(X
):IF T(Y)<>C-1 THEN 1270
1240 IF Y=F(X) GOSUB 1330:GOTO
1270
1250 IF Y<>SE THEN Y=S(Y):GOTO
1240
1260 Y=F(X):S(Y)=S(X):F(S(Y))=Y
```

```
:T(Y)=C:FE=Y:LA=C
1270 NEXT A
1280 NEXT C:PRINT"EVENTO INICIA
L";U(SE);", EVENTO FINAL ";U(FE
)
1290 FOR E=1 TO EE:Y=E(E)
1300 IF F(Y)=0 AND Y<>FE THEN P
RINT U(Y);"NAO CONECTADO AO EVE
NTO FINAL":CK=FA
1310 NEXT E:IF CK THEN 1370
1320 GOTO 1540
1330 CLS:CK=FA:PRINT"EXISTE O S
EGUINTE LOOP":PRINT"EVENTOS ...
":XA=A(A)
1340 X=F(XA):PRINT U(X):Y=S(XA)
:PRINT U(Y)
1350 Y=S(Y):PRINT U(Y):IF Y<>X
THEN 1350
1360 FOR X=1 TO 1000:NEXT:RETUR
N
1370 K=1:AK=AA:IF AA=1 THEN K=0
1380 AK=INT((AK+K)/2):IF AK=0 T
HEN 1430
1390 K=0:FOR A=AK+1 TO AA:B=A-A
K:X=A(A):Y=A(B):XE=S(X):YE=S(Y)
1400 IF T(YE)+YE/ZZ<=XE/ZZ+T(XE
) THEN 1420
1410 A(A)=Y:A(B)=X:K=1
1420 NEXT A:GOTO 1380
1430 N(FE)=LA:FOR D=LA-1 TO 1 S
TEP-1
1440 FOR A=1 TO AA:X=A(A):IF N(
F(X))<>D+1 THEN 1460
1450 Y=S(X):F(Y)=F(X):N(Y)=D
1460 NEXT A:NEXT D
1470 FOR A=1 TO AA:G(A)=A(A):NE
XT A:K=1:AK=AA:IF AA=1 THEN K=0
1480 AK=INT((AK+K)/2):IF AK=0 T
HEN 1530
1490 K=0: FOR A=AK+1 TO AA:B=A-
AK:X=G(A):Y=G(B):XE=F(X):YE=F(Y
)
1500 IF N(YE)+YE/ZZ<=XE/ZZ+N(XE
) THEN 1520
1510 G(B)=X:G(A)=Y:K=1
1520 NEXT A:GOTO 1480
1530 CK=TR:RETURN
1540 CK=FA:FOR X=1 TO 1000:NEXT
 X:RETURN
1550 GOSUB 1870:FOR A=1 TO AA:X
=A(A):Z(X)=T(X):NEXT A:GOSUB 16
20
1560 FOR A=1 TO AA:X=A(A):Y(X)=
-(Z(F(X))-Y(S(X))=Z(X))*100.NEX
T A
1570 FOR B=1 TO AA STEP 3:CLS:F
OR A=B TO AA+FNA(B+2-AA):X=A(A)
1580 PRINT#PR,A$;U(X);"=";U$(X)
1590 C=Y(S(X)):D=Z(F(X)):PRINT
#PR,"PODE INICIAR";C;"DEVE TERM
INAR";D
1600 PRINT#PR,"TEMPO LIVRE";INT
(100*(D-C-Z(X)))/100;"(CRITICO"
;Y(X);"%)":IF T=9 THEN PRINT#PR
,USING"DESVIO=####.##";Q(X)
1610 PRINT#PR:NEXT A:GOSUB 1020
:NEXT B:RETURN
1620 FOR E=1 TO EE:Y(E(E))=0:NE
XT E
1630 FOR A=1 TO AA:X=A(A):Y(F(X
))=Y(F(X))+FNZ(Y(S(X))-Y(F(X))+
Z(X)):NEXT A
1640 FOR E=1 TO EE:Z(E(E))=Y(FE
):NEXT E:FOR A=AA TO 1 STEP-1:X
=G(A)
1650 Z(S(X))=Z(S(X))+FNA(Z(F(X)
)-Z(S(X))-Z(X)):NEXT A:RETURN
1660 GOSUB 1870:FOR A=1 TO AA:X
=A(A):P(X)=0:Q(X)=0:Y(X)=0:NEXT
 A
1670 FOR E=1 TO EE:Z=E(E):P(Z)=
0:Q(Z)=0:NEXT E
1680 FOR M=1 TO 43 STEP 3:FOR A
=1 TO AA:W(A)=2*RND(0)-1:NEXT A
1690 FOR N=0 TO 4 STEP 2:CLS:PR
INT"CASO";M+N/2;" EM 45"
1700 FOR A=1 TO AA:X=A(A):TX=T(
X):IF TX=0 THEN Z(X)=0:GOTO 175
0
1710 NX=N(X):IF NX=TX THEN Z(X)
=TX:GOTO 1750
1720 W=FNW(W(A)+N/3):IF NX>=TX*
3 THEN Z(X)=-NX*(W<TX/NX):GOTO
1750
1730 IF NX>TX*2.34 THEN Z(X)=-T
X*LOG(W):GOTO 1750
1740 W=FNX(W-.5):Z(X)=ABS(TX+W*
(NX-TX))
1750 NEXT A
1760 GOSUB 1620
1770 FOR A=1 TO AA:X=A(A):Z=Z(F
(X))-Y(S(X))-Z(X)
1780 P(X)=P(X)+Z:Q(X)=Q(X)+Z*Z:
Y(X)=Y(X)+(Z<1E-6):NEXT A
1790 FOR E=1 TO EE:Z=E(E):P(Z)=
P(Z)+Y(Z):Q(Z)=Q(Z)+Z(Z):NEXT E
,N,M
1800 FOR E=1 TO EE:Z=E(E):Y(Z)=
VAL(LEFT$(STR$(P(Z)/45),6))
1810 Z(Z)=VAL(LEFT$(STR$(Q(Z)/4
5),6)):NEXT E
1820 FOR A=1 TO AA:X=A(A):Y=Y(X
):Y(X)=-VAL(LEFT$(STR$(Y/45*100
),4))
1830 IF P(X)<1E-2 THEN P(X)=0
1840 Z=45-Y+.1E-9:Z(X)=Z(F(X))-
Y(S(X))-VAL(LEFT$(STR$(P(X)/Z),
6))
1850 Q(X)=SQR(ABS((Q(X)-P(X)*P(
X)/Z)/((Z-1)+.1E-9))):IF Q(X)<1
.E-6 THEN Q(X)=0
1860 NEXT A:GOTO 1570
1870 IF(PEEK(65314)AND1)=1 THEN
 RETURN ELSE CLS:PRINT"TELA OU
IMPRESSORA (T/I) ?"
1880 Q$=INKEY$:IF Q$<>"T" AND Q
$<>"I" THEN 1880
1890 IF Q$="I" THEN PR=-2
1900 CLS:RETURN
```

```
600  HOME : PRINT DOS"OPEN "F$
605  PRINT DOS"READ "F$
610  INPUT MA,ME,MH,AA,EE,CK: G
OSUB 12
615  IF CK THEN  INPUT SE,FE
620  FOR A = 1 TO AA
625  INPUT X,U(X),S(X),F(X),T(X
),N(X),G(A),U$(X):A(A) = X: NEX
T A
630  FOR E = 1 TO EE
635  INPUT X,U(X),S(X),F(X),T(X
),N(X),U$(X):E(E) = X: NEXT E
640  INPUT X: IF X > 0 THEN U(X
) = ZZ + 1: GOTO 640
645  PRINT DOS"CLOSE "F$
650  RETURN
700  PRINT DOS"OPEN "F$
710  PRINT DOS"DELETE "F$
720  PRINT DOS"CLOSE "F$
730  RETURN
800  GOSUB 942
810  FOR A = 1 TO AA:X = A(A):
GOSUB 932
820 Y = Y + 1 + ( LEN (U$(X)) >
 12): IF Y > 20 AND (A < AA) TH
EN  GOSUB 940: GOSUB 942
830  NEXT A: GOSUB 940: PRINT "
EVENTOS":Y = 3
840  FOR E = 1 TO EE:X = E(E):X
P = U(X): GOSUB 950: PRINT U$(X
)
850 Y = Y + 1 + ( LEN (U$(X)) >
 12): IF Y > 20 AND E < EE THEN
  GOSUB 940: GOSUB 946
860  NEXT E: GOTO 940
900  INPUT "REINICIA O PROGRAMA
 (S/N)? ";ANS: IF  LEFT$ (ANS,1
) = "N" THEN 50
910  IF ANS <  > "S" THEN 900
920  RUN
932 XP =  FN U(S(X)): GOSUB 950
:XP =  FN U(F(X)): GOSUB 950:XP
 = T(X): GOSUB 950:XP = N(X)
933  GOSUB 950:XP =  ABS (U(X))
: GOSUB 950
935  PRINT : PRINT "TEXTO = ";U
$(X): RETURN
940  IF KK$ <  > "S" THEN  PRIN
T "<RETURN> PARA CONTINUAR";: I
NPUT F$: HOME : RETURN
941  RETURN
942  HOME : PRINT "ATIVIDADES:"
943  PRINT "--EVENTOS-- ---TEMP
0--- ------"
944  PRINT "FINAL INICI PROVL E
STIM CODIGO":Y = 3: RETURN
950 XP$ =  LEFT$ ( STR$ (XP) +
"      ",6): PRINT XP$;: RETURN

960  HOME : PRINT "SAIDA P/ IMP
RESSORA (S/N)?"
970  GET KK$: IF KK$ <  > "N" A
ND KK$ <  > "S" THEN 970
980  RETURN
1000 CK = TR: FOR A = 1 TO AA:X
 = A(A)
1020 XE = S(X): IF S(XE) < 0 OR
 ZZ < U(XE) THEN  PRINT U(X);W$
(5);U(XE):CK = FA
1030 Z = F(X): IF S(Z) < 0 OR Z
Z < U(Z) THEN  PRINT U(X);W$(5)
;U(Z):CK = FA
1040  NEXT A: IF (CK = FA) THEN
 1750
1050 E = 1
1060 X = E(E): IF S(X) < 0 THEN
  GOSUB 400: IF E <  = EE THEN
1060
1070 E = E + 1: IF E <  = EE TH
EN 1060
1080  FOR E = 1 TO EE:X = E(E):
S(X) = 0:F(X) = 0: NEXT E
1082  FOR A = 1 TO AA:X = A(A):
S(F(X)) = X: NEXT A
1090 SE = 0: FOR E = 1 TO EE:X
= E(E): IF S(X) > 0 THEN 1096
```

```
1092  IF SE = 0 THEN SE = X: GO
TO 1096
1094  PRINT WS(1);U(X): IF SE <
  = MH THEN  PRINT WS(1);U(SE):
SE = MH + 1
1096  NEXT E: IF SE = 0 THEN  P
RINT "TODOS EVENTOS TEM ";A$;"
PRECEDENDO"
1098  IF SE = 0 OR (SE > MH) TH
EN 1750
1100  FOR E = 1 TO EE:X = E(E):
T(X) = 0:N(X) = 0: NEXT E:T(SE)
 = 1
1110 LA = 1: FOR C = 2 TO EE +
2: IF LA <  > C - 1 THEN 1170
1120  FOR A = 1 TO AA:X = A(A):
Y = S(X): IF T(Y) <  > C - 1 TH
EN 1160
1130  IF Y = F(X) THEN  GOSUB 1
200: GOTO 1160
1140  IF (Y <  > SE) THEN Y = S
(Y): GOTO 1130
1150 Y = F(X):S(Y) = S(X):F(S(Y
)) = Y:T(Y) = C:FE = Y:LA = C
1160  NEXT A
1170  NEXT C: PRINT "EVENTO INI
CIAL =";U(SE);", EVENTO FINAL =
";U(FE)
1180  FOR E = 1 TO EE:Y = E(E)
1190  IF F(Y) = 0 AND (Y <  > F
E) THEN  PRINT U(Y);"NAO CONECT
ADO AO EVENTO FINAL":CK = FA
1192  NEXT E: IF CK THEN 1300
1194  GOTO 1750
1200  HOME : PRINT "EXISTE O SE
GUINTE LOOP ": PRINT "EVENTOS..
.":XA = A(A)
1210 X = F(XA): PRINT U(X):Y =
S(XA): PRINT U(Y)
1220 Y = S(Y): PRINT U(Y): IF Y
 <  > X THEN 1220
1230  RETURN
1300 K = 1:AK = AA: IF AA = 1 T
HEN K = 0
1310 AK =  INT ((AK + K) / 2):
IF AK = 0 THEN 1500
1320 K = 0: FOR A = AK + 1 TO A
A:B = A - AK:X = A(A):Y = A(B):
XE = S(X):YE = S(Y)
1330  IF T(YE) + YE / ZZ <  = T
(XE) + XE / ZZ THEN 1360
1340 A(A) = Y:A(B) = X:K = 1
1360  NEXT A: GOTO 1310
1500 N(FE) = LA: FOR D = LA - 1
 TO 1 STEP  - 1
1520  FOR A = 1 TO AA:X = A(A):
 IF N(F(X)) <  > D + 1 THEN 156
0
1550 Y = S(X):F(Y) = F(X):N(Y)
= D
1560  NEXT A,D
1600  FOR A = 1 TO AA:G(A) = A(
A): NEXT A:K = 1:AK = AA: IF AA
 = 1 THEN K = 0
1610 AK =  INT ((AK + K) / 2):
IF AK = 0 THEN 1700
1620 K = 0: FOR A = AK + 1 TO A
A:B = A - AK:X = G(A):Y = G(B):
XE = F(X):YE = F(Y)
1630  IF N(YE) + YE / ZZ <  = N
(XE) + XE / ZZ THEN 1660
1650 G(B) = X:G(A) = Y:K = 1
1660  NEXT A: GOTO 1610
1700 CK = TR: RETURN
1750 CK = FA: INPUT "TECLE <RET
URN>";HG$: RETURN
2000  FOR A = 1 TO AA:X = A(A):
Z(X) = T(X): NEXT A: GOSUB 2100
2020  FOR A = 1 TO AA:X = A(A):
Y(X) = (Z(F(X)) - Y(S(X)) = Z(X
)) * 100: NEXT A: GOSUB 2100
2030  FOR B = 1 TO AA STEP 5: H
OME : FOR A = B TO AA +  FN A(B
 + 4 - AA):X = A(A)
2040  PRINT : PRINT A$;U(X);" =
";U$(X)
2050 C = Y(S(X)):D = Z(F(X))
2055  PRINT "PODE INICIAR "; IN
T (C * 100) / 100;", DEVE TERMI
NAR " INT (D * 100) / 100
2060  PRINT "TEMPO LIVRE = "; I
NT ((D - C - Z(X)) * 100) / 100
;"  ("; INT (Y(X) * 100) / 100;
"% CRITICO)"
2065  IF T = 12 THEN  PRINT "DE
SVIO = "; INT (Q(X) * 100) / 10
0;
2070  PRINT : NEXT A: GOSUB 940
: NEXT B: RETURN
2100  FOR E = 1 TO EE:Y(E(E)) =
 0: NEXT E
2110  FOR A = 1 TO AA:X = A(A):
Y(F(X)) = Y(F(X)) +  FN Z(Y(S(X
)) - Y(F(X)) + Z(X)): NEXT A
2120  FOR E = 1 TO EE:Z(E(E)) =
 Y(FE): NEXT E: FOR A = AA TO 1
 STEP  - 1:X = G(A)
2130 Z(S(X)) = Z(S(X)) +  FN A(
Z(F(X)) - Z(S(X)) - Z(X)): NEXT
 A: RETURN
3000  FOR A = 1 TO AA:X = A(A):
P(X) = 0:Q(X) = 0:Y(X) = 0: NEX
T A
3020  FOR E = 1 TO EE:X = E(E):
P(X) = 0:Q(X) = 0: NEXT E
3030  FOR M = 1 TO 43 STEP 3: F
OR A = 1 TO AA:W(A) = 2 *  RND
(1) - 1: NEXT A
3040  FOR N = 0 TO 4 STEP 2: HO
ME : PRINT "CASO #";M + N / 2;"
/45"
3050  FOR A = 1 TO AA:X = A(A):
TX = T(X): IF TX = 0 THEN Z(X)
= 0: GOTO 3080
3052 NX = N(X): IF (NX = TX) TH
EN Z(X) = TX: GOTO 3080
3054 W =  FN W(W(A) + N / 3): I
F NX >  = TX * 3 THEN Z(X) = NX
 * (W < TX / NX): GOTO 3080
3060  IF NX > TX * 2.34 THEN Z(
X) =  - TX *  LOG (W): GOTO 308
0
3070 W =  FN X(W - .5):Z(X) =
ABS (TX + W * (NX - TX))
3080  NEXT A
3090  GOSUB 2100
3100  FOR A = 1 TO AA:X = A(A):
Z = Z(F(X)) - Y(S(X)) - Z(X)
3110 P(X) = P(X) + Z:Q(X) = Q(X
) + Z * Z:Y(X) = Y(X) - (Z < 1.
E - 6): NEXT A
3120  FOR E = 1 TO EE:X = E(E):
P(X) = P(X) + Y(X):Q(X) = Q(X)
+ Z(X): NEXT E,N,M
3125  IF KK$ = "S" THEN  PRINT
DOS"PR# 1"
3200  FOR E = 1 TO EE:X = E(E):
Y(X) =  VAL ( LEFT$ ( STR$ (P(X
) / 45),6))
3210 Z(X) =  VAL ( LEFT$ ( STR$
 (Q(X) / 45),6)): NEXT E
3220  FOR A = 1 TO AA:X = A(A):
Y = Y(X):Y(X) =  - VAL ( LEFT$
 ( STR$ (Y / 45 * 100),4))
3230  IF P(X) < 1.E - 2 THEN P(
X) = 0
3240 Z = (45 - Y) + .1E - 9:Z(X
) = Z(F(X)) - Y(S(X)) - VAL (
LEFT$ ( STR$ (P(X) / Z),6))
3250 Q(X) =  SQR ((Q(X) - P(X)
* P(X) / Z) / ((Z - 1) + .1E -
9)): IF Q(X) < 1.E - 6 THEN Q(X
) = 0
3260  NEXT A: GOTO 2030
```



```
590 IF F$<>A$ THEN 680
600 FOR B=1 TO AA:IF X=A(B) THE
N A=B
610 NEXT B:A(A)=A(AA):U(X)=ZZ+1
:AA=AA-1:RETURN
620 IF U(X)<0 GOSUB 1050:PRINT
USING(#####  ";ABS(U(X));:PRINT
 U$(X):GOTO 650
630 IF EE=ME THEN PRINT W$(Z);F
$:RETURN
640 GOSUB 660
650 PRINT W$(4):F$:INPUT U$(X):
S(X)=0:RETURN
660 EE=EE+1:E(EE)=X:S(X)=-1:F(X
)=0:U(X)=U
670 T(X)=0:N(X)=0:U$(X)="":RETU
RN
680 Z=X:FOR F=1 TO EE:IF E(F)=Z
 THEN E=F
690 NEXT F:E(E)=E(EE):U(Z)=ZZ+1
:EE=EE-1:RETURN
700 Z=U-INT((U-1)/MH)*MH:Y=Z:X=
0
710 IF X=0 (0=U(Z) OR ZZ+1=U(Z)
)THEN X=Z
720 IF U=U(Z) THEN X=Z:RETURN
730 IF Y=1 OR 0=U(Z) THEN RETUR
N
740 Z=Z+Y-MH*INT((Z+Y-1)/MH):Y=
Y+Y-MH*INT((Y+Y-1)/MH):GOTO 170
750 OPEN "CAS:"+F$ FOR OUTPUT A
S #3
755 PRINT#3, MAZ$MEZ$MHZ$AAZ$EE
Z$EEZ$CKZ$
760 IF CK THEN PRINT#3,SEZ$FE
770 FOR A=1 TO AA:X=A(A):PRINT
#3,X;Z$U(X)Z$S(X)Z$F(X)Z$T(X)Z$
N(X)Z$G(A)Z$U$(X):NEXT A
780 FOR E=1 TO EE:Z=E(E):PRINT#
3,Z;Z$U(Z)Z$S(Z)Z$F(Z)Z$T(Z)Z$N
(Z)Z$U$(Z):NEXT E
790 FOR X=1 TO MH:IF U(X)=ZZ+1
THEN PRINT #3,X
800 BEXT X:PRINT#3,0
810 CLOSE#3:MOTORON:FOR X=1 TO
100:NEXT X:MOTOROFF:RETURN
820 CLS:PRINT"SALVAR DADOS - er
ro:FOR K=1 TO 1000:NEXT:RETURN
830 OPEN "CAS:"+F$ FOR INPUT AS
 #3:INPUT #3,MA,ME,MH,AA,EE,CK:
GOSUB 20
```

```
840 IF CK THEN INPUT #3,SE,FE
850 FOR A=1 TO AA:INPUT #3,X,U(
X),S(X),F(X),T(X),N(X),G(A),U$(
X):A(A)=X:NEXT A
860 FOR E=1 TO EE:INPUT#3,Z,U(Z
),S(Z),F(Z),T(Z),N(Z),U$(Z):E(E
)=Z:NEXT E
870 INPUT #3,X:IF X>0 THEN U(X)
=ZZ+1:GOTO 870
880 CLOSE #3:RETURN
890 GOSUB 1870:A=0:GOSUB 1030
900 FOR A=1 TO AA:X=A(A):GOSUB
1000
910 Y=Y+1:IF Y>8 AND A<AA GOSUB
 1020:GOSUB 1030
920 NEXT A:GOSUB 1020
930 E=0:GOSUB 1050:FOR E=1 TO E
E:X=E(E):PRINT #PR,USING"#####
  ";ABS(U(X));:PRINT #PR,U$(X)
940 Y=Y+1:IF Y>15 AND E<EE GOSU
B 1020:GOSUB 1050
950 NEXT E:GOTO 1020
960 CLS:PRINT"TEM CERTEZA ?  (S
/N)"
970 T$=INKEY$:IF T$<>"S" AND T$
<>"N" THEN 970
980 IF T$="N" THEN RETURN
990 CLS:END
1000 PRINT #PR,USING"##### ####
# ##### #####  #####";FNU(S(X))
,FNU(F(X)),T(X),N(X),ABS(U(X));
:IF PR=0 THEN PRINT
1010 PRINT #PR," ";U$(X):RETURN
1020 IF PR=1 THEN PRINT"<ENTER>
 PARA CONTINUAR":INPUT F$:CLS:R
ETURN ELSE RETURN
1030 CLS:IF PR=1 OR A=0 THEN PR
INT #PR,"INICIO ULTIMA VEZ   90
% ATIVIDADE":Y=3
1040 RETURN
1050 CLS:IF PR=1 OR E=0 THEN PR
INT #PR," EVENTO  TEXTO":Y=3
1060 RETURN
1070 CK=TR:FOR A=1 TO AA:X=A(A)
1080 Z=S(X):IF S(Z)<0 OR ZZ<U(Z
) THEN PRINT A$;U(X);W$(5);U(Z)
:CK=FA
1090 Z=F(X):IF S(Z)<0 OR ZZ<U(Z
) THEN PRINT A$;U(X);W$(5);U(Z)
:CK=FA
1100 NEXT A:IF CK=FA THEN 1540
1110 E=1
1120 Z=E(E):IF S(Z)<0 GOSUB 680
:IF E<=EE THEN 1120
1130 E=E+1:IF E<=EE THEN 1120
1140 FOR E=1 TO EE:Z=E(E):S(Z)=
0:F(Z)=0:NEXT E
1150 FOR A=1 TO AA:X=A(A):S(F(X
))=X:NEXT A
1160 SE=0:FOR E=1 TO EE:Z=E(E):
IF S(Z)>0 THEN 1190
1170 IF SE=0 THEN SE=Z:GOTO 119
0
1180 PRINT W$(1);U(Z):IF SE<=MH
 THEN PRINT W$(1);U(SE):SE=MH+1
1190 NEXT E:IF SE=0 THEN PRINT"
TODOS OS EVENTOS TEM":PRINT A$;
"PRECEDENDO"
1200 IF SE=0 OR SE>MH THEN 1540
1210 FOR E=1 TO EE:Z=E(E):T(Z)=
0:N(Z)=0:NEXT E:T(SE)=1
1220 LA=1:FOR C=2 TO EE+2:IF LA
<>C-1 THEN 1280
1230 FOR A=1 TO AA:X=A(A):Y=S(X
):IF T(Y)<>C-1 THEN 1270
1240 IF Y=F(X) GOSUB 1330:GOTO
1270
1250 IF Y<>SE THEN Y=S(Y):GOTO
1240
1260 Y=F(X):S(Y)=S(X):F(S(Y))=Y
:T(Y)=C:FE=Y:LA=C
1270 NEXT A
1280 NEXT C:PRINT"EVENTO INICIA
L";U(SE);", EVENTO FINAL ";U(FE
)
1290 FOR E=1 TO EE:Y=E(E)
1300 IF F(Y)=0 AND Y<>FE THEN P
RINT U(Y);"NAO CONECTADO AO EVE
NTO FINAL":CK=FA
1310 NEXT E:IF CK THEN 1370
1320 GOTO 1540
1330 CLS:CK=FA:PRINT"EXISTE O S
EGUINTE LOOP":PRINT"EVENTOS ...
":XA=A(A)
1340 X=F(XA):PRINT U(X):Y=S(XA)
:PRINT U(Y)
1350 Y=S(Y):PRINT U(Y):IF Y<>X
THEN 1350
1360 FOR X=1 TO 1000:NEXT:RETUR
N
1370 K=1:AK=AA:IF AA=1 THEN K=0
1380 AK=INT((AK+K)/2):IF AK=0 T
HEN 1430
1390 K=0:FOR A=AK+1 TO AA:B=A-A
K:X=A(A):Y=A(B):XE=S(X):YE=S(Y)
1400 IF T(YE)+YE/ZZ<=XE/ZZ+T(XE
) THEN 1420
1410 A(A)=Y:A(B)=X:K=1
1420 NEXT A:GOTO 1380
1430 N(FE)=LA:FOR D=LA-1 TO 1 S
TEP-1
1440 FOR A=1 TO AA:X=A(A):IF N(
F(X))<>D+1 THEN 1460
1450 Y=S(X):F(Y)=F(X):N(Y)=D
1460 NEXT A:NEXT D
1470 FOR A=1 TO AA:G(A)=A(A):NE
XT A:K=1:AK=AA:IF AA=1 THEN K=0
1480 AK=INT((AK+K)/2):IF AK=0 T
HEN 1530
1490 K=0: FOR A=AK+1 TO AA:B=A-
AK:X=G(A):Y=G(B):XE=F(X):YE=F(Y
)
1500 IF N(YE)+YE/ZZ<=XE/ZZ+N(XE
) THEN 1520
1510 G(B)=X:G(A)=Y:K=1
1520 NEXT A:GOTO 1480
1530 CK=TR:RETURN
1540 CK=FA:FOR X=1 TO 1000:NEXT
 X:RETURN
1550 GOSUB 1870:FOR A=1 TO AA:X
=A(A):Z(X)=T(X):NEXT A:GOSUB 16
20
1560 FOR A=1 TO AA:X=A(A):Y(X)=
-(Z(F(X))-Y(S(X))=Z(X))*100:NEX
T A
1570 FOR B=1 TO AA STEP 3:CLS:F
OR A=B TO AA+FNA(B+2-AA):X=A(A)
1580 PRINT#PR,A$;U(X);"=";U$(X)
1590 C=Y(S(X)):D=Z(F(X)):PRINT
#PR,"PODE INICIAR";C;"DEVE TERM
INAR";D
1600 PRINT#PR,"TEMPO LIVRE";INT
(100*(D-C-Z(X)))/100;"(CRITICO"
;Y(X);"%)":IF T=9 THEN PRINT#PR
,USING"DESVIO=####.##";Q(X)
1610 PRINT#PR:NEXT A:GOSUB 1020
:NEXT B:RETURN
1620 FOR E=1 TO EE:Y(E(E))=0:NE
XT E
1630 FOR A=1 TO AA:X=A(A):Y(F(X
))=Y(F(X))+FNZ(Y(S(X))-Y(F(X))+
Z(X)):NEXT A
1640 FOR E=1 TO EE:Z(E(E))=Y(FE
):NEXT E:FOR A=AA TO 1 STEP-1:X
=G(A)
1650 Z(S(X))=Z(S(X))+FNA(Z(F(X)
)-Z(S(X))-Z(X)):NEXT A:RETURN
1660 GOSUB 1870:FOR A=1 TO AA:X
=A(A):P(X)=0:Q(X)=0:Y(X)=0:NEXT
 A
1670 FOR E=1 TO EE:Z=E(E):P(Z)=
0:Q(Z)=0:NEXT E
1680 FOR M=1 TO 43 STEP 3:FOR A
=1 TO AA:W(A)=2*RND(0)-1:NEXT A
1690 FOR N=0 TO 4 STEP 2:CLS:PR
INT"CASO";M+N/2;" EM 45"
1700 FOR A=1 TO AA:X=A(A):TX=T(
X):IF TX=0 THEN Z(X)=0:GOTO 175
0
1710 NX=N(X):IF NX=TX THEN Z(X)
=TX:GOTO 1750
1720 W=FNW(W(A)+N/3):IF NX>=TX*
3 THEN Z(X)=-NX*(W<TX/NX):GOTO
1750
1730 IF NX>TX*2.34 THEN Z(X)=-T
X*LOG(W):GOTO 1750
1740 W=FNX(W-.5):Z(X)=ABS(TX+W*
(NX-TX))
1750 NEXT A
1760 GOSUB 1620
1770 FOR A=1 TO AA:X=A(A):Z=Z(F
(X))-Y(S(X))-Z(X)
1780 P(X)=P(X)+Z:Q(X)=Q(X)+Z*Z:
Y(X)=Y(X)+(Z<1E-6):NEXT A
1790 FOR E=1 TO EE:Z=E(E):P(Z)=
P(Z)+Y(Z):Q(Z)=Q(Z)+Z(Z):NEXT E
,N,M
1800 FOR E=1 TO EE:Z=E(E):Y(Z)=
VAL(LEFT$(STR$(P(Z)/45),6))
1810 Z(Z)=VAL(LEFT$(STR$(Q(Z)/4
5),6)):NEXT E
1820 FOR A=1 TO AA:X=A(A):Y=Y(X
):Y(X)=-VAL(LEFT$(STR$(Y/45*100
),4))
1830 IF P(X)<1E-2 THEN P(X)=0
1840 Z=45-Y+.1E-9:Z(X)=Z(F(X))-
Y(S(X))-VAL(LEFT$(STR$(P(X)/Z),
6))
1850 Q(X)=SQR(ABS((Q(X)-P(X)*P(
X)/Z)/((Z-1)+.1E-9))):IF Q(X)<1
.E-6 THEN Q(X)=0
1860 NEXT A:GOTO 1570
1870 CLS:PRINT"TELA OU IMPRESSO
RA (T/I) ?"
1880 Q$=INKEY$:IF Q$<>"T" AND Q
$<>"I" THEN 1880
1890 IF Q$="I" THEN PR=2
1900 CLS:RETURN
```

Se você tem um acionador de disquete acoplado ao seu MSX, proceda às seguintes modificações:

```
750 OPEN "A:"+F$ FOR OUTPUT AS
#3
830 OPEN "A:"+F$ FOR INPUT AS#3
:INPUT#3,MA,ME,MH,AA,EE,CK:GOSU
B 20
```

LINHA	FABRICANTE	MODELO
Apple II +	Appletronica	Thor 2010
Apple II +	CCE	MC-4000 Exato
Apple II +	CPA	Absolutus
Apple II +	CPA	Polaris
Apple II +	Digitus	DGT-AP
Apple II +	Dismac	D-8100
Apple II +	ENIAC	ENIAC II
Apple II +	Franklin	Franklin
Apple II +	Houston	Houston AP
Apple II +	Magnex	DM II
Apple II +	Maxitronica	MX-2001
Apple II +	Maxitronica	MX-48
Apple II +	Maxitronica	MX-64
Apple II +	Maxitronica	Maxitronic I
Apple II +	Microcraft	Craft II Plus
Apple II +	Milmar	Apple II Plus
Apple II +	Milmar	Apple Master
Apple II +	Milmar	Apple Senior
Apple II +	Omega	MC-400
Apple II +	Polymax	Maxxi
Apple II +	Polymax	Poly Plus
Apple II +	Spectrum	Microengenho I
Apple II +	Spectrum	Spectrum ed
Apple II +	Suporte	Venus II
Apple II +	Sycomig	SIC I
Apple II +	Unitron	AP II
Apple II +	Victor do Brasil	Elppa II Plus
Apple II +	Victor do Brasil	Elppa Jr.
Apple IIe	Microcraft	Craft IIe
Apple IIe	Microdigital	TK-3000 IIe
Apple IIe	Spectrum	Microengenho II
MSX	Gradiente	Expert GPC-1
MSX	Sharp	Hotbit HB-8000
Sinclair Spectrum	Microdigital	TK-90X
Sinclair Spectrum	Timex	Timex 2000
Sinclair ZX-81	Apply	Apply 300
Sinclair ZX-81	Engebras	AS-1000
Sinclair ZX-81	Filcres	NEZ-8000
Sinclair ZX-81	Microdigital	TK-82C
Sinclair ZX-81	Microdigital	TK-83
Sinclair ZX-81	Microdigital	TK-85
Sinclair ZX-81	Prologica	CP-200
Sinclair ZX-81	Ritas	Ringo R-470
Sinclair ZX-81	Timex	Timex 1000
Sinclair ZX-81	Timex	Timex 1500
TRS-80 Mod. I	Dismac	D-8000
TRS-80 Mod. I	Dismac	D-8001/2
TRS-80 Mod. I	LNW	LNW-80
TRS-80 Mod. I	Video Genie	Video Genie I
TRS-80 Mod.III	Digitus	DGT-100
TRS-80 Mod.III	Digitus	DGT-1000
TRS-80 Mod.III	Kemitron	Naja 800
TRS-80 Mod.III	Prologica	CP-300
TRS-80 Mod.III	Prologica	CP-500
TRS-80 Mod.III	Sysdata	Sysdata III
TRS-80 Mod.III	Sysdata	Sysdata Jr.
TRS-80 Mod.IV	Multix	MX-Compacto
TRS-80 Mod.IV	Sysdata	Sysdata IV
TRS-Color	Codimex	CS-6508
TRS-Color	Dynacom	MX-1600
TRS-Color	LZ	Color 64
TRS-Color	Microdigital	TKS-800
TRS-Color	Prologica	CP-400

FABRICANTE	MODELO	PAÍS	LINHA
Appletronica	Thor 2010	Brasil	Apple II +
Apply	Apply 300	Brasil	Sinclair ZX-81
CCE	MC-4000 Exato	Brasil	Apple II +
CPA	Absolutus	Brasil	Apple II +
CPA	Polaris	Brasil	Apple II +
Codimex	CS-6508	Brasil	TRS-Color
Digitus	DGT-100	Brasil	TRS-80 Mod.III
Digitus	DGT-1000	Brasil	TRS-80 Mod.III
Digitus	DGT-AP	Brasil	Apple II +
Dismac	D-8000	Brasil	TRS-80 Mod. I
Dismac	D-8001/2	Brasil	TRS-80 Mod. I
Dismac	D-8100	Brasil	Apple II +
Dynacom	MX-1600	Brasil	TRS-Color
ENIAC	ENIAC II	Brasil	Apple II +
Engebras	AS-1000	Brasil	Sinclair ZX-81
Filcres	NEZ-8000	Brasil	Sinclair ZX-81
Franklin	Franklin	USA	Apple II +
Gradiente	Expert GPC1	Brasil	MSX
Houston	Houston AP	Brasil	Apple II +
Kemitron	Naja 800	Brasil	TRS-80 Mod.III
LNW	LNW-80	USA	TRS-80 Mod. I
LZ	Color 64	Brasil	TRS-Color
Magnex	DM II	Brasil	Apple II +
Maxitronica	MX-2001	Brasil	Apple II +
Maxitronica	MX-48	Brasil	Apple II +
Maxitronica	MX-64	Brasil	Apple II +
Maxitronica	Maxitronic I	Brasil	Apple II +
Microcraft	Craft II Plus	Brasil	Apple II +
Microcraft	Craft IIe	Brasil	Apple IIe
Microdigital	TK-3000 IIe	Brasil	Apple IIe
Microdigital	TK-82C	Brasil	Sinclair ZX-81
Microdigital	TK-83	Brasil	Sinclair ZX-81
Microdigital	TK-85	Brasil	Sinclair ZX-81
Microdigital	TK-90X	Brasil	Sinclair Spectrum
Microdigital	TKS-800	Brasil	TRS-Color
Milmar	Apple II Plus	Brasil	Apple II +
Milmar	Apple Master	Brasil	Apple II +
Milmar	Apple Senior	Brasil	Apple II +
Multix	MX-Compacto	Brasil	TRS-80 Mod.IV
Omega	MC-400	Brasil	Apple II +
Polymax	Maxxi	Brasil	Apple II +
Polymax	Poly Plus	Brasil	Apple II +
Prologica	CP-200	Brasil	Sinclair ZX-81
Prologica	CP-300	Brasil	TRS-80 Mod.III
Prologica	CP-400	Brasil	TRS-Color
Prologica	CP-500	Brasil	TRS-80 Mod.III
Ritas	Ringo R-470	Brasil	Sinclair ZX-81
Sharp	Hotbit HB-8000	Brasil	MSX
Spectrum	Microengenho I	Brasil	Apple II +
Spectrum	Microengenho II	Brasil	Apple IIe
Spectrum	Spectrum ed	Brasil	Apple II +
Suporte	Venus II	Brasil	Apple II +
Sycomig	SIC I	Brasil	Apple II +
Sysdata	Sysdata III	Brasil	TRS-80 Mod.III
Sysdata	Sysdata IV	Brasil	TRS-80 Mod.IV
Sysdata	Sysdata Jr.	Brasil	TRS-80 Mod.III
Timex	Timex 1000	USA	Sinclair ZX-81
Timex	Timex 1500	USA	Sinclair ZX-81
Timex	Timex 2000	USA	Sinclair Spectrum
Unitron	AP II	Brasil	Apple II +
Victor do Brasil	Elppa II Plus	Brasil	Apple II +
Victor do Brasil	Elppa Jr.	Brasil	Apple II +
Video Genie	Video Genie I	USA	TRS-80 Mod. I

INPUT foi especialmente projetado para microcomputadores compatíveis com as sete principais linhas existentes no mercado.
Os blocos de textos e listagens de programas aplicados apenas a determinadas linhas de micros podem ser identificados por meio dos seguintes símbolos:

Quando o emblema for seguido de uma faixa, então tanto o texto como os programas que se seguem passam a ser específicos para a linha indicada.

NO PRÓXIMO NÚMERO

APLICAÇÕES
Um programa de referência cruzada. Como utilizar. Aplicações.

SOFTWARE
Sistemas de gerenciamento de bancos de dados. Importação e exportação. Chaves de acesso e de ordenação. Saídas.

PERIFÉRICOS
Processamento de imagens: câmara de vídeo e dispositivos CCD.

SUMÁRIO GERAL
Relação dos artigos, por títulos, das várias seções de ***INPUT***.

SUMÁRIO DOS QUADROS
Temas Gerais, Microdicas, Perguntas e Respostas, Tabelas.